Technical University of Cartagena

Department of Information Technologies and Communications

Advances in analytical models and applications for RFID, WSN and AmI systems

PhD THESIS

Francisco Javier Parrado García

2015

Advances in analytical models and applications for RFID, WSN and AmI systems

Francisco Javier Parrado García

Advisor

Javier Vales Alonso

Technical University of Cartagena

Department of Information Technologies and Communications

2015

Universidad Politécnica de Cartagena

DT-16

CONFORMIDAD DE SOLICITUD DEAUTORIZACIÓN DE DEPÓSITO DE TESIS DOCTORAL POR EL/LA DIRECTOR/A DE LA TESIS

D. **Javier Vales Alonso** Director de la Tesis doctoral **"Advances in analytical models and applications for RFID, WSN and AmI systems"**.

INFORMA:

Que la referida Tesis Doctoral, ha sido realizada por D. **Francisco Javier Parrado García**, dentro del programa de doctorado Tecnologías de la Información y Comunicaciones, dando mi conformidad para que sea presentada ante la Comisión de Doctorado para ser autorizado su depósito.

La rama de conocimiento en la que esta tesis ha sido desarrollada es:

- ☐ Ciencias
- ☐ Ciencias Sociales y Jurídicas
- ☒ Ingeniería y Arquitectura

En Cartagena, a 16 de febrero de 2015

EL DIRECTOR DE LA TESIS

NOMBRE VALES ALONSO JAVIER - NIF [redacted]

Firmado digitalmente por NOMBRE VALES ALONSO JAVIER - [redacted]
Nombre de reconocimiento (DN): c=ES, o=FNMT, ou=FNMT Clase 2 CA, ou=500053084, cn=NOMBRE VALES ALONSO JAVIER - N[redacted]
Fecha: 2015.02.16 16:39:00 Z

Fdo.: Javier Vales Alonso

COMISIÓN DE DOCTORADO

Universidad Politécnica de Cartagena

DT-17

CONFORMIDAD DE DEPÓSITO DE TESIS DOCTORAL POR LA COMISIÓN ACADÉMICA DEL PROGRAMA

D. **Fernando Daniel Quesada Pereira**, Presidente de la Comisión Académica del Programa Tecnologías de la Información y las Comunicaciones.

INFORMA:

Que la Tesis Doctoral titulada, "**Advances in Analytical Models and Applications for RFID, WSN, and AmI Systems**", ha sido realizada, dentro del mencionado programa de doctorado, por **D. Francisco Javier Parrado García**, bajo la dirección y supervisión del **Dr. Javier Vales Alonso**.

En reunión de la Comisión Académica de fecha **16/02/2015**, visto que en la misma se acreditan los indicios de calidad correspondientes y la autorización del Director de la misma, se acordó dar la conformidad, con la finalidad de que sea autorizado su depósito por la Comisión de Doctorado.

La Rama de conocimiento por la que esta tesis ha sido desarrollada es:

- ☐ Ciencias
- ☐ Ciencias Sociales y Jurídicas
- ☒ Ingeniería y Arquitectura

En Cartagena, a 18 de Febrero de 2015

EL PRESIDENTE DE LA COMISIÓN ACADÉMICA DEL PROGRAMA

NOMBRE QUESADA PEREIRA FERNANDO DANIEL - NIF [redacted]

Firmado digitalmente por NOMBRE QUESADA PEREIRA FERNANDO DANIEL - NIF [redacted]
Nombre de reconocimiento (DN): c=ES, o=FNMT, ou=FNMT Clase 2 CA, ou=500890415, cn=NOMBRE QUESADA PEREIRA FERNANDO DANIEL - NIF [redacted]
Fecha: 2015.02.18 14:39:32 +01'00'

Fdo: Fernando D. Quesada Pereira

COMISIÓN DE DOCTORADO

He remembers those vanished years.
As though looking through a dusty window pane,
the past is something he could see, but not touch.
And everything he sees is blurred and indistinct.
That era has passed.
Nothing that belonged to it exists anymore.

Acknowledgements

The harder you try to forget something, the more it will stick in your memory. Once I heard someone say that if you have to lose something, the best way is to keep it in your memory.

I would like to express my sincere thanks to everyone who contributed to this research, directly or indirectly. Above all, special thanks to my advisor Javier for his guidance and support within this work. A small fraction of the gratitude toward him is about this thesis, the most part of it is about the opportunity he has given me to learn. And last, but not least, I am very much indebted to my family whom I owe everything, what I have and what I am.

Abstract

– What is it? Artificial intelligence?
– Incorrect. I am not AI. I am a living, thinking entity that was created in the sea of information.

Internet of Things (IoT) is being built upon many different elements acting as sources and sinks of information, rather than the previous human-centric Internet conception. Developments in IoT include a vast set of fields ranging from data sensing, to development of new protocols and applications. Indeed, a key concept underlying in the conception of IoT is the smart and autonomous processing of the new huge data flows available. In this work, we aim to study three different aspects within IoT.

First, we will focus on the sensing infrastructure. Among the different kind of sensing technologies available to IoT systems, **Radio Frequency Identification (RFID)** is widely considered one of the leading technologies. RFID is the enabling technology behind applications such as access control, tracking and tracing of containers, file management, baggage sorting or equipment location. With the grow up of RFID, many facilities require multiple RFID readers usually operating close to each other. These are known as Dense Reader Environments (DREs). The co-existence of several readers operating concurrently is known to cause severe interferences on the identification process. One of the key aspects to solve in RFID DREs is achieving proper coordination among readers. This is the focus of the first part of this doctoral thesis.

Unlike previous works based on heuristics, we address this problem through an optimization-based approach. The goal is identifying the maximum mean number of tags while network constraints are met. To be able to formulate these optimization problems, we have obtained analytically the mean number of identifications in a bounded -discrete or continuous- time period, an additional novel contribution of our work. Results show that our approach is overwhelmingly better than previous known methods.

Along sensing technologies of IoT, **Wireless Sensor Networks (WSNs)** plays a fundamental role. WSNs have been largely and theoretically studied in the past decade, and many of their initial problems related to communication aspects have been successfully solved. However, with the adoption of WSNs in real-life projects, new issues have arisen, being one of them the development of realistic strategies to deploy WSNs. We have studied different ways of solving this aspect by focusing on different optimality criteria and evaluating the different trade-offs that occur when a balanced solution must be selected.

On the one hand, deterministic placements subject to conflicting goals have been addressed. Results can be obtained in the form of Pareto-frontiers, allowing proper solution selection. On the other hand, a number of situations correspond to deployments were the nodes' position is inherently random. We have analyzed these situations leading first to a theoretical model, which later has been particularized to a Moon WSN survey. Our work is the first considering a full model with realistic properties such as 3D topography, propellant consumptions or network lifetime and mass limitations.

Furthermore, development of smart applications within IoT is the focus of the **Ambient Intelligence (AmI)** field. Rather than having people adapting to the surrounding environment, AmI pursues the development of sensitive environments able to anticipate support in people's actions. AmI is progressively being introduced in many real-life environments like education, homes, health and so forth. In this thesis we develop a sport-oriented AmI system designed to improve athletes training. The goal is developing an assistant able to provide real-time training orders based on both environment and athletes' biometry, which is aimed to control the aerobic and the technical-tactical training. Validation experiments with the honor league UCAM Volleyball Murcia team have shown the suitability of this approach.

Resumen

– You talk about redefining my identity, I want a guarantee that I can still be myself.

– There isn't one. Why would you wish to? All things change in a dynamic environment. Your effort to remain what you are is what limits you.

Internet de las cosas (IoT) está constituido por equipos y elementos que actúan como fuentes y sumideros de información, a diferencia de la percepción clásica de Internet, centrado en las personas. Los avances en IoT engloban un amplio número de áreas y tecnologías, desde la adquisición de información hasta el desarrollo de nuevos protocolos y aplicaciones. Un concepto clave que subyace en el concepto de IoT, es el procesamiento inteligente y autónomo de los flujos de información que se dispone. En este trabajo, estudiamos tres aspectos diferentes de IoT.

En primer lugar, nos centraremos en la infraestructura de obtención de datos. Entre las diferentes tecnologías de obtención de datos disponibles en los sistemas IoT, la **Identificación por Radio Frecuencia (RFID)** es considerada como una de las tecnologías predominantes. RFID es la tecnología detrás de aplicaciones tales como control de acceso, seguimiento y trazabilidad de contenedores, gestión de archivos, clasificación de equipaje o localización de equipos. Con el auge de la tecnología RFID, muchas instalaciones empiezan a requerir la presencia de múltiples lectores RFID que operan próximos entre sí y conjuntamente. A estos escenarios se les conoce como *Dense Reader Environments* (DREs). La coexistencia de varios lectores operando simultáneamente puede causar graves problemas de interferencias en el proceso de identificación. Uno de los aspectos claves a resolver en los RFID DREs consiste en lograr la coordinación adecuada entre los lectores. Estos problemas de coordinación se tratan con detalle en esta tesis doctoral.

A diferencia de trabajos anteriores donde se proponen soluciones basadas en heurísticos, en esta tesis se proponen soluciones basadas en modelos de optimización. El objetivo es identificar el número máximo de *tags*, dadas una serie de restricciones en la red de lectores. Para poder formular este tipo de problemas de optimización, se han obtenido analíticamente las ecuaciones que permiten calcular el número medio de identificaciones en un período de tiempo acotado, de naturaleza continua y discreta. Este análisis es otra contribución novedosa de esta tesis. Los resultados demuestran que nuestro enfoque, basado en optimización, es muy superior a los métodos heurísticos desarrollados hasta la fecha.

Además, dentro del área de obtención de datos relativa a IoT, las **Redes de Sensores Inalámbricas (WSNs)** desempeñan un papel fundamental. Durante la última década, las WSNs han sido estudiadas ampliamente de forma teórica, y la mayoría de problemas relacionados con la comunicación en este tipo de redes se han conseguido resolver de forma favorable. Sin embargo, con la implementación de WSNs en proyectos reales, han surgido nuevos problemas, siendo uno de ellos el desarrollo de estrategias realistas para desplegar las WSN. En este trabajo se estudian diferentes métodos que abordan esta cuestión, centrándonos en distintos criterios de optimización, y analizando las diferentes ventajas e inconvenientes que se producen al buscar una solución equilibrada.

Por una parte se han abordado despliegues de sensores de tipo determinista sujetos a optimización multiobjetivo, donde los resultados óptimos constituyen una frontera de Pareto. Por otra parte, se han estudiado casos en los que los nodos se despliegan de manera aleatoria. Se ha obtenido en primer lugar un modelo teórico, que posteriormente se ha particularizado en un caso concreto de despliegue de WSNs sobre la superficie lunar. Este trabajo es el primero en abordar un modelo completo y realista de despliegue aleatorio de sensores que considera características tales como topografías 3D, consumos de combustible, o limitaciones en el tiempo de vida de la red o en la masa.

Por último, la **Inteligencia Ambiental (AmI)** forma parte del desarrollo de aplicaciones inteligentes en IoT. Hasta ahora, han sido las personas quienes han tenido que adaptarse al entorno. En cambio, AmI persigue crear entornos capaces de anticipar y apoyar las acciones de las personas. AmI se está introduciendo progresivamente en diversos entornos reales tales como la educación, la salud, domótica, etc. En esta tesis se introduce un sistema AmI orientado al deporte que busca mejorar el entrenamiento de los deportistas. El objetivo prioritario es el desarrollo de un asistente capaz de proporcionar órdenes de entrenamiento en tiempo real, basadas tanto en el entorno como en parámetros biométricos de los atletas. Experimentos llevados a cabo con el equipo de división de honor UCAM Volleyball Murcia han demostrado la viabilidad de este asistente.

Contents

Acknowledgements ix

Abstract xi

Introduction 1

I Optimization in RFID dense reader environments 7

1. Performance analysis of optimal schedulers in single channel DREs 11
1.1. Introduction . . . 11
1.2. Related work . . . 12
1.2.1. Tag identification procedure . . . 14
1.3. Optimal time distribution . . . 15
1.3.1. $\varphi(n_j, t_j)$ computation for Static-FSA . . . 17
1.3.2. $\varphi(n_j, t_j)$ computation for Dynamic-FSA . . . 18
1.4. Results . . . 18
1.5. Tag estimation impact on scheduler performance . . . 28
1.5.1. Numerical examples . . . 29
1.6. Conclusions . . . 30
Appendix A. Computation of $P(a|n, t)$ . . . 32

2. An Optimization-based Scheduler for RFID DREs 35
2.1. Introduction . . . 35
2.2. Related work . . . 36
2.3. FSA tag identification procedure . . . 37
2.3.1. $\varphi(n_j, t_j)$ computation for SFSA . . . 38
2.3.2. $\varphi(n_j, t_j)$ computation for DFSA . . . 38
2.4. OSL - Optimization-based Scheduler with Load information . . . 39
2.4.1. Normal Form . . . 40
2.4.2. Compact Form . . . 41

2.5. OSL implementation 42
2.5.1. Simulated annealing solver 42
2.5.2. Convergence analysis 44
2.5.3. Practical use of OSL 46
2.5.4. Synchronization issues 47
2.5.5. Dense reader environments with mobile readers 47
2.6. Results 47
2.6.1. Comparison of OSL *versus* heuristics 48
2.6.2. OSL performance analysis 51
2.6.3. Performance in DREs with mobile readers 54
2.7. Conclusions 54

3. Mean number of tag identifications in FSA during a time interval 55
3.1. Introduction 55
3.2. System model 58
3.3. Identification performance analysis 59
3.4. Results 61
3.5. Conclusions 64

II Wireless Sensor Networks Deployment 65

4. Acustic Sensor Planning for Gunshot Location in National Parks 69
4.1. Introduction 69
4.2. Gunshot location 71
4.2.1. System architecture 72
4.2.2. Acoustic source location procedure 72
4.2.3. Synchronization schema 75
4.3. Optimization model 76
4.4. Solving the optimization model 78
4.4.1. Alternative approaches 78
4.4.2. Derivative-free unconstrained minimization 79
4.5. Example of deployment 80
4.5.1. Conclusions 85

5. Random deployment of WSN in non-homogeneous scenarios 89
5.1. Introduction 89
5.2. Related work 91
5.3. Sensor Positioning Model 94
5.4. Optimization model 96

5.4.1. Single cluster analysis 99
5.4.2. Statement of the global optimization 104
5.4.3. Statement of the iterative optimization 105
5.5. Deployment examples and results 106
5.5.1. Validation tests 107
5.5.2. Global versus Iterative solution 110
5.5.3. An iterative optimization example: The Tycho crater 111
5.5.4. Iterative optimization versus heuristic placement 114
5.6. Conclusions 115

6. Optimal planning of WSN deployments for in situ lunar surveys 117
6.1. Introduction 117
6.1.1. Mission background 118
6.1.2. Work contributions 120
6.2. Related works 120
6.3. Lunar mission concept 123
6.4. Sensing-connectivity model 126
6.4.1. Sensor ejection mechanism 127
6.4.2. Cluster connectivity 128
6.4.3. $\mathbb{E}\{\Gamma\}$ computation 132
6.4.4. Sensing-connectivity balance examples 133
6.5. Mass model 133
6.5.1. $\boldsymbol{\Delta v}$ budget computation 136
6.6. Energy model 138
6.7. Formal optimization problem statement 139
6.8. Optimization problem solving 140
6.8.1. Solution generation algorithm 141
6.9. Optimal deployments in the Bullialdus crater 142
6.10. Conclusions 149

III AmI system for personalized sport training 151

7. A smart coaching assistant for professional volley training 155
7.1. Introduction 155
7.2. Related work 157
7.2.1. Physical performance in sports 157
7.2.2. AmI systems in the sport domain 158
7.3. System architecture 159
7.4. Effort control module 161

7.4.1. Decision-making process 162
7.4.2. Best-fit polynomial approximation 163
7.4.3. Kalman filter 164
7.4.4. Sliding-Window distribution estimation 165
7.4.5. Validation 167
7.5. Technical-tactical control system 172
7.5.1. Dynamic Programming model 174
7.5.2. Technical-tactical effort control module 177
7.6. Exercise quality control module 182
7.6.1. Jump detection using k-NN classifier 184
7.6.2. Jump intensity analysis 185
7.7. Conclusions 187

Conclusions and future work **189**
Optimization in RFID dense-reader environments 189
WSN Deployment 190
AmI system for personalized sport training 191

Bibliography **193**

List of Figures

1.0. Interferences in dense reader environments 8

1.1. Example scenarios for $m = 6$ 19
1.2. Expected number of identifications (Φ) versus n. Full-mesh scenario 22
1.3. Expected number of identifications (Φ) versus n. Star scenario . . . 23
1.4. Expected number of identifications (Φ) versus n. Line scenario . . . 24
1.5. Optimal vs proportional allocation for Full-mesh topology 25
1.6. Optimal vs proportional allocation for Star topology 26
1.7. Optimal vs proportional allocation for Line topology 27

2.1. Example of scheduling in a RFID dense reader environment with f=2 39
2.2. Annealing algorithm convergence speed 44
2.3. Annealing algorithm speed performance 45
2.4. Example scenarios 48
2.5. Optimal *vs.* proportional allocation, Full-mesh scenario 49
2.6. Graphical representations for the optimal solutions in the 6R scenario 50
2.7. Expected number of identifications (Φ) vs. n. Star scenario 52
2.8. Expected number of identifications (Φ) vs. n. Full-mesh scenario . . 53

3.1. Durations of different command sequences in an ISO 18000-6C frame computed from the timing parameters 56
3.2. Reading process model 58
3.3. Mean number of identifications versus time, SFSA k=16 61
3.4. Mean number of identifications versus time, DFSA 62
3.5. Mean number of identifications versus time, n=20 62
3.6. Throughput versus time 63
3.7. SFSA optimal frame-length configuration versus n and t 64

4.1. Landscape of the Cabañeros National Park 70
4.2. Gunshot location architecture 72
4.3. Hyperbolic multilateration of an acoustic signal 73

4.4. Regression line performed in a level k node, with several broadcast time stamps from a level $k-1$ node 76
4.5. Synthetic outdoor scenario for the numerical tests 81
4.6. Aggregated distance to the power lines (cost) vs. coverage area with $V(x) \geq 1$ 83
4.7. Aggregated distance to the power lines (cost) vs. coverage area with $V(x) \geq 2$ 83
4.8. Aggregated distance to the power lines (cost) vs. coverage area with $V(x) \geq 3$ 84
4.9. A sensor deployment for $\theta = 0.1$ 85
4.10. A sensor deployment for $\theta = 0.5$ 86
4.11. A sensor deployment for $\theta = 0.9$ 87
4.12. Percentage of covered grid points versus number of detecting sensors for different values of θ 87

5.1. System model 95
5.2. Effect of sensor dispersion in the importance covered. Importance is captured only in the circled areas 101
5.3. $\frac{\Omega}{N}$ curves 102
5.4. LROC WAC image of Tycho crater 108
5.5. Analytic vs experimental multi-deployment Tycho map 109
5.6. Analytic versus experimental importance 109
5.7. Multi-deployment solution, global method, homogeneous map 110
5.8. Multi-deployment solution for Tycho map 112
5.9. Final multi-deployment solution for Tycho map 113
5.10. Joint sensing coverage $\Gamma(\%)$ versus M, Tycho map 113
5.11. Cell division for the heuristic placement, Tycho map 114

6.1. Bullialdus Crater 119
6.1. Overall sensors deployment strategy 125
6.2. Deployment sequence 128
6.3. Effect of sensor dispersion in the importance covered varying σ . . . 133
6.4. Effect of sensor dispersion in the importance covered at fixed σ and varying P_{tx} 134
6.5. Optimal solution, $P_{\text{tx}} = 0$ dBm, $\rho = 0.01$ 144
6.6. Optimal solution, $P_{\text{tx}} = -20$ dBm, $\rho = 0.01$ 145
6.7. Optimal solution, $P_{\text{tx}} = 0$ dBm, $\rho = 0.1$ 146
6.8. Optimal solution, $P_{\text{tx}} = -20$ dBm, $\rho = 0.1$ 147
6.9. Coverage versus payload limit P 148
6.10. Trade-off QFD diagram 150

7.0. General vision of ambient intelligence aimed at Sports 152

7.1. Global system architecture 160
7.2. SAETA biometric hardware 161
7.3. SAETA main window 162
7.4. Input window for customized match module 163
7.5. Kalman filter cycle 166
7.6. Real-time HR samples during a customized match training 168
7.7. Real samples of hitter heart rate and their distribution in intensity classes 169
7.8. Estimation error 170
7.9. Final Δ *versus* STEP value, player 1 171
7.10. Final Δ *versus* STEP value, player 2 172
7.11. UCAM Volley Murcia hitter repetition exercises 173
7.12. Graphical representation of Bellman´s equation 174
7.13. Evolution of the HR of a player during a training session as a function of time 177
7.14. Configuration scheme of a technical-tactical training session 179
7.15. Total value function J^0 181
7.16. Technical-tactical module input and output windows 182
7.17. Different jump actions 183
7.18. Partial record of labelled jumping/non-jumping data during training sessions 185
7.19. Peak X acceleration value for valid and not valid jump actions . . . 186
7.20. Quality module input and output windows 186

List of Tables

1.1. Full-mesh scenario. Optimal assignment of slots for Dynamic-FSA . 20
1.2. Star scenario. Optimal assignment of slots for Dynamic-FSA 20
1.3. Line scenario. Optimal assignment of slots for Dynamic-FSA 20
1.4. Full-mesh scenario. Ratio of improvement using tag estimation . . . 29
1.5. Star scenario. Ratio of improvement using tag estimation 30
1.6. Line scenario. Ratio of improvement using tag estimation 30
1.7. Full-mesh scenario. Ratio of improvement using tag estimation . . . 31
1.8. Star scenario. Ratio of improvement using tag estimation 31
1.9. Line scenario. Ratio of improvement using tag estimation 31

4.1. Non-monotone derivative-free algorithm 80
4.2. Gunshot average location error (m) 86

5.1. $\frac{\Omega}{N}$ fitting coefficients 103
5.2. Analytical vs experimental values computed averaging 5000 experiments 107
5.3. Performance of iterative vs global methods 111
5.4. Improvement in terms of covered importance of the optimal solution compared with the heuristic 115

6.1. Table caption text 129
6.2. Payload description of Lunar missions 134
6.3. Table caption text 139
6.4. Rules for generation of a new solution 143

7.1. Possible values for the control parameter u_n 176
7.2. Examples of optimal policies for HR ranges 183
7.3. True positives for the classifier for several k values and feature sets (F) 185

Introduction

If we all reacted the same way, we'd be predictable, and there's always more than one way to view a situation. What's true for the group is also true for the individual. It's simple: Overspecialize, and you breed in weakness. It's slow death.

The concept of the Internet of Things (IoT) first became popular in 1999 [Ashton, 2009] and it was closely related to Radio Frequency Identification (RFID). RFID was described as a prerequisite for the IoT in those days: if all objects and people in daily life were equipped with identifiers, computers could manage and inventory them [Magrassi and Berg, 2002].

[Ashton, 2009] claims that today, computers and Internet are almost completely dependent on human beings for information. Nearly all of data available on the Internet were first captured and created by human beings (typing, pressing a record button, taking a digital picture or scanning a bar code). The problem is that people have limited time, attention and accuracy, all of which means they are not good enough at capturing data about things in the real world. RFID and sensor technology enable computers to observe, identify and understand the world without the limitations of human-entered data.

The involvement of intelligence was not implicit in this original concept of IoT (just tagging objects and retrieving data of them from Internet). However, over the years, the early concept of IoT has evolved due to the convergence of multiple technologies: Internet, wireless communications or embedded systems among others. [Miorandi et al., 2012] states that in terms of enabling technologies, a key issue for IoT is the development of appropriate means for identifying smart objects and enabling interactions with the environment. In this sense, key building blocks are expected to be represented by Wireless Sensor Networks (WSNs) and RFID. As far as WSNs are concerned, the ability of sensing the environment and to self-organize into ad hoc networks represents important features from an IoT perspective.

In close relation with the IoT in the last years, the new concept of Ambient Intelligence (AmI) has surfaced. AmI refers to potential fields where the system supports intelligently the activities, reacting to environment changes as well as to user actions. AmI is focused on the application of artificial intelligence methods in IoT environments, together with environments rich in sensing, computing and actuation capabilities, which are designed to respond in an intelligent way to the presence of users, supporting them in carrying out specific tasks.

Therefore, among other fields, RFID, WSNs and AmI play a relevant role to enable the IoT, even with possibilities beyond its original conception.

This thesis proposes several contributions to the aforementioned fields. These contributions are mainly based on the application of optimization and analysis methods. Next we provide a short introduction to the different fields and our specific contributions.

Optimization in RFID dense reader environments

The first part of this work discusses a specific scenario within passive RFID technology called *RFID dense reader environments*. RFID systems are one of the enabling technologies for the ubiquitous computing paradigm. At the moment, the EPCglobal organization leads the development of industry-driven standards for this field and has settled the EPC Gen 2 [EPCglobal, 2013] as a reference standard.

RFID technology is intended to replace barcode systems, allowing a reliable and fast identification, even with difficult electro-magnetic environmental conditions or without line-of-sight between reader and object to be identified. This technology is increasingly being used to identify and trace objects, for example, in supply chains, industrial manufacturing and traceability processes. These environments are characterized by a large number of items with attached tags which flow on conveyor belts, inside pallets or boxes, and the like, entering and leaving facilities. In this kind of large realistic installations, multiple readers are operating in proximity of one another. In technical terms, this is known as a *RFID dense reader environment* (DRE).

The co-existence of several RFID readers operating concurrently to identify *tags* attached to the objects, may adversely affect the performance because readers transmitting simultaneously easily overpower the dim tags' responses, ruining the identification process. To avoid these interferences, the readers must transmit either at different times or in different frequency channels. Therefore, some coordination mechanism must establish a suitable scheduling among the readers.

The scheduling problem for generic single-channel environments, and an extended model to the most general (arbitrary number of readers, time slots and topology of the network) multi-channel case are presented in Chapters 1 and 2 respectively. Both were defined as Mixed Integer Non-Linear Programming (MINLP)

problems. The former is solved by means of the General Algebraic Modeling System (GAMS) [Brooke et al., 1998] (a high-level modeling system for mathematical programming and optimization), and a solver based on the Extended Cutting Plane (AlphaECP) [Westerlund, 1992]. In the latter we present an efficient solution generation mechanism for a parallel version of Simulated Annealing (SA) [Kirkpatrick et al., 1983] meta-heuristic.

Furthermore, the mathematical development of the aforementioned problems, requires to compute the number of identifications that will take place in RFID systems within an identification interval. Most characterizations of the RFID reading process, deal with the computation of the mean population identification time or the number of identifications per interrogation round. However, although connected, they are different problems, and we have developed the proper analytical formulations to obtain these statistics. Two models are studied:

(I) In a first step, a *discrete* time model is addressed. The aim is at computing the identifications that will take place within a number of available time slots.

(II) After this characterization, the analysis of a *continuous* model, where the identifications within an interval of t seconds are computed. Although this model is more complex than the previous one, it allows a more realistic characterization of the RFID reading process (see Chapter 3).

From this knowledge we are able to formulate suitable schedulers for RFID DREs.

The work relative to this part has led to different publications. Contributions reported in Chapter 1 are supported by the publications [Vales-Alonso et al., 2012b, 2013d]. Results in Chapter 2 are under review, whereas Chapter 3 has been published in [Vales-Alonso et al., 2014].

WSNs deployment

The second part of this thesis addresses a number of problems related to WSNs deployments and the study of the trade-offs involved. WSNs offer new capabilities to study large-scale physical world phenomena, leading to unprecedented observing abilities. A WSN consist of a network of spatially distributed and autonomous sensors that are able to cooperatively deliver their sensed data to a main location or node, the *sink*. WSNs are proved to be a useful tool in multitude of applications, including environmental sensing (*e.g.* air pollution or water quality monitoring, natural disaster prevention), industrial monitoring or even health care supervision among others.

In order to deploy the sensing nodes of a WSN, in some scenarios the position of the nodes can be accurately selected, whereas in other scenarios, the only suitable option for deployment is a random scattering of the nodes (*e.g.* dropping sensors

from a plane over difficult to access or high risk areas). The former is a *deterministic* placement, while the latter is a *random* deployment.

This thesis aims at developing mathematical models and meta-heuristics to select the optimal deployment of the nodes, considering trade-offs between cost, quality of sensor data and wireless transmission reliability, in both *deterministic* and *random* deployments.

An example of deterministic network placement scenario related to poaching detection is exposed in Chapter 4. Using a derivative-free descent methodology, the aim is to obtain the placement configuration with maximal network sensing coverage and minimal cost. Since both goals are contrary, a Pareto front must be considered.

In chapters related to random deployment, a more analytical and complex problem is proposed, where we assume that Gaussian distributed nodes are forming several clusters. In a first approach (see Chapter 5), a simplified model is discussed: the deployment takes place over a plain surface and we assume that nodes are grouped in several clusters given a predetermined number of nodes and clusters. The goal of this model is to decide the launch point and dispersion for each cluster in order to maximize the gathered information, fulfilling certain *connectivity* and *coverage* constraints. The optimization algorithm has been implemented using GAMS, and solved with CONOPT [Drud, 1985], a solver for large-scale Non-Linear Problems (NLP) optimization.

After this simplified model, a thorough analytical and realistic deployment model over the lunar surface is discussed in Chapter 6. In this model, the roughness of the surface is considered and the connectivity between nodes is calculated through a realistic radio propagation model. Furthermore, the number of nodes and clusters are not prefixed, but depend on factors like: maximum mass that can be placed in a lunar spacecraft, propellant consumption or sensors battery capacity required to operate during a full lunar cycle. To tackle this problem, we have to rely on a SA algorithm with a specific solution generation mechanism where a Monte Carlo method is present.

Research conducted in this part, have resulted in the following publications: Chapter 4 is based on [González-Castaño et al., 2009], Chapter 5 corresponds to the publication [Vales-Alonso et al., 2013b]. Results in Chapter 6 are under review.

AmI for personalized sport training

Finally, the last part of this thesis is devoted to developing Ambient Intelligence (AmI) systems oriented to sport training. Until recently, people have continuously adapted to their surrounding environment in order to survive and improve their quality of life. However, AmI systems have the potential to create intelligent envi-

ronments that are sensitive, intelligently responsive and able to anticipate actions to the presence of people with the aim to support them. AmI examples include spaces for education, smart homes, health, sports, leisure, transportation, and so forth.

In Chapter 7 we introduce SAETA, a sport-oriented AmI system for athletes training, in particular, a smart assistant for professional volleyball training based on supervised learning techniques. This system relies on a sensing infrastructure able to monitor environment conditions (*e.g.* temperature or humidity) as well as athletes biometry and making decisions during a training session to meet not only the training goals (*e.g.* maximizing effort while avoiding fatigue), but the proper execution of the exercises.

Real-time data gathered is analyzed by a decision engine which is divided into three modules: *effort control module*, that estimates the ratios of each effort regime, based on three different methodologies (best-fit polynomial approximations, Kalman filter [Kalman, 1960] and Sliding-Window distribution estimation); *technical-tactical control module*, based on a Dynamic Programming (DP) [Bellman and Dreyfus, 1962] model, which selects the best activity and resting durations in repetitional training with the goal of maximizing effort while avoiding fatigue; and the *exercise quality control module*, which controls the quality of the motions by evaluating their intensities by means of a k-Nearest Neighbors (k-NN) classifier [Fix and Hodges Jr, 1951].

Project SAETA has resulted in a number of publications. In particular, Chapter 7 is based on articles [Parrado-García et al., 2014] and [Vales-Alonso et al., 2015].

Part I

Optimization in RFID dense reader environments

– Slow down! You know if I lose my grip on you I'll fall through this cloud!

– Well if you would stop thinking bad thoughts then it wouldn't be a problem!

Passive Radio Frequency Identification (RFID) is increasingly being used to identify and trace objects in supply chains [Gaukler, 2011], [Ondemir et al., 2012], industrial manufacturing and traceability processes [Nguyen et al., 2013], [Wu et al., 2013], smart environments and Internet of Things [Xu et al., 2014], accurate location [Pomarico-Franquiz and Shmaliy, 2014], [Abu-Mahfouz and Hancke, 2013], and so forth. These environments are characterized by a large number of items with attached tags which flow on conveyor belts, inside pallets or boxes, and the like, entering and leaving facilities. In large realistic installations several readers are commonly deployed, these are the so-called *dense reader environments* (DREs), which comprise multiple readers operating concurrently.

In these facilities readers' output power may reach up to a few watts to achieve a reader-to-tag range of several meters, but causing indirecly interferences to other readers, which may be of two different kinds.

On the one hand, Reader-to-Tag Interferences (RTI) occur when readers share a common reading area. For example, in Figure 1.0, if readers R_1 and R_2 power some tags simultaneously (e.g. tag A), these tags are not able to produce any response. This type of interferences are usually addressed by separating the reading zones,

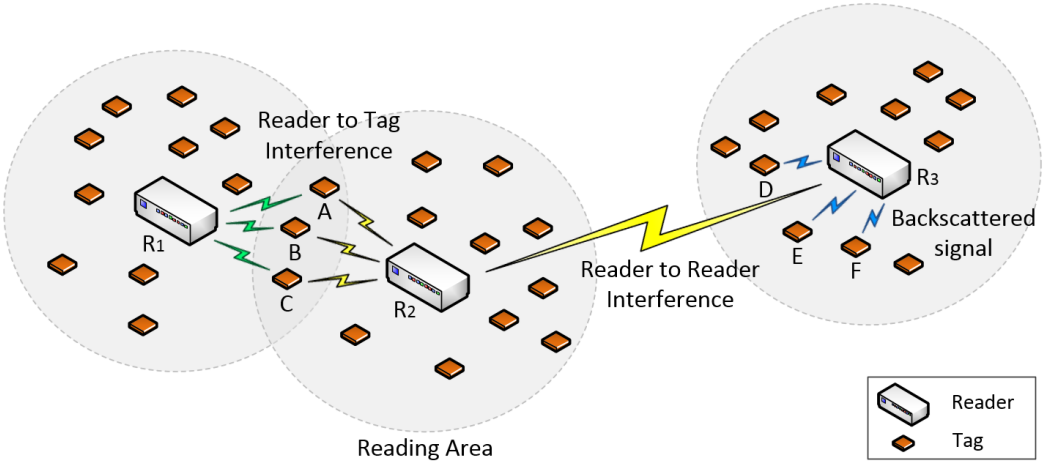


Figure 1.0: Interferences in dense reader environments

either physically or with a proper scenario arrangement that may involve changing the aiming of the antennas, screening using for the readers or other options.

On the other hand, Reader-to-Reader Interferences (RRI) occur when one reader outpowers the extremely low power tag's response to other reader. For example, in Figure 1.0, tags within reading area of R_3 can not be read if at the same time R_2 operates and *vice versa*. To put into perspective, the signal of a reader transmitting from distances of kilometers is still received with more power than the response of a tag located at a few meters away. Of course a proper scenario arrangement helps to mitigate the problem, but RRI problem may remain. In these cases, it is required that both readers and tags transmit at different times or at different frequency channels. This leads to a problem of collisions among readers, and this part of the thesis is devoted to addressing it.

Related works usually deal with this problem by means of distributed algorithms of the Listen-Before-Talk (LBT) family like Colorwave, DCS, DCNS (see [Gandino et al., 2013] for a reference about these protocols) or NFRA [Eom et al., 2009]. These proposals operate selecting a "color" for each reader. Each scheduling period is divided into time slot blocks, each one associated to a particular color. Readers with a given color can transmit using only the corresponding blocks. The final performance depends on the number of colors, which algorithms adjust dynamically using different criteria, and "solving" the collisions on the process, *i.e.* how colors are reassigned after a collision event. Besides, another issue of these heuristics is that some "colors" (*i.e.* time blocks) may remain unused if no reader selects them. Currently standards propose either LBT-like approaches like the aforementioned works, or the use of tags capable of being multiplexed in frequency with the readers. However, the latter option is at the expense of a reduced communication bandwidth, thus limiting the reading speed.

In this thesis we assume that tags do not support multiplex operation, as the main works in the field, but we propose a different methodology to address the collisions. We assume that basic information about the reading process is available to a central coordinator which uses it to decide periodically the best *schedule* for all the readers, that is, which reader must transmit at a given time and which channel it must use.

From our perspective, a number of parameters in the scenario are relevant to find the best scheduling, such as the number of readers and their topology (i.e. which pairs have RRI interferences), the length of the scheduling period, the available number of channels, and so forth. Besides, the goal of the scheduler is achieve optimal performance in the network, in terms of total identifications per scheduling period.

The driven idea in our work plan is to characterize the expected performance in RFID networks, and afterwards propose schedulers based on this information. The first part is done by analytically determining the expected number of identifications when n tags contend in a given time period, either specified in time slots or in seconds. Although many RFID performance characterizations exists, they have traditionally being aimed at computing the total time to identify a population. This approach is not longer valid for RFID schedulers since in this case the important performance figure is the number of identifications in the allocated time.

Chapters 1 and 2 deal with the *discrete-time* case, where time period is specified in time slots. In these Chapters, this characterization is used to propose schedulers based on Mixed Linear Integer Non-Convex programs, for single-channel and multiple-channel scenarios, respectively.

The analysis of the *continuous* model, where the identifications within an interval of t seconds are computed is provided in Chapter 3. This scenario is more complex than the previous one, but it allows a more realistic characterization of the RFID reading process. From this analysis we expect to derive also mathematical programs leading to optimal scheduler formulations in our forthcoming works.

Chapter 1

Performance analysis of optimal schedulers in single channel dense RFID environments

Do you know what people did in the old days when they had secrets they didn't want to share? They'd climb a moutain, find a tree, carve a hole in it, whisper the secret into the hole and cover it up with mud. That way, nobody else would ever learn the secret...

1.1. Introduction

The scheduling problem for single-channel environments is introduced in this Chapter. That is, we consider the case of any arbitrary number of readers (m) and for any particular network topology. Attending to the interference relative restrictions mentioned in the introduction, in this case, the readers cannot transmit simultaneously if reader-to-reader interferences are present.

In addition, we provide insight on the impact in the schedulers derived from the knowledge of the tag population associated to each reader. To the best of our knowledge, all previous optimization models (see Section 1.2) have largely ignored the availability of this information. This information can be effectively exploited to construct a scheduler with the goal of maximizing the number of identifications in the whole interrogation period. In this work, we assume that this information is known, and show how it can be used to develop an optimal scheduler. Moreover, we analyze the improvement obtained when this information is available, and the effect on the expected performance when errors occur in tag estimation

This resource allocation problem is addressed both for static and dynamic frame length identification procedures (which are described later in Section 1.2.1), and that the tags remain in coverage of their corresponding reader at least during the whole period of identification (t time slots). The goal is maximizing the expected number of identified tags in the whole network.

The rest of the Chapter is organized as follows: In Section 1.2 the most relevant research proposals are shown. Section 1.2.1 describes the identification procedures commonly used in RFID readers. Section 1.3 describes the optimization model. Section 1.4 shows the performance results achieved by the optimal algorithm. Section 1.5 deals with the analysis of the impact of tag population estimation in the scheduler. Finally, Section 1.6 concludes and describes future works.

1.2. Related work

A number of proposal for coordinating dense reader environments have been presented in the literature, most of them are based on heuristic approaches and are, thus, suboptimal by nature. A summary of these works is contained in [Bueno-Delgado et al., 2010].

Besides, a number of papers [Choi and Lee, 2011; Kim et al., 2007; Chui-Yu et al., 2009; Deolalikar et al., 2005; Mohsenian-Rad et al., 2010; Tanaka and Sasase, 2007; Seo and Lee, 2010] propose different system models, and schedulers based on the optimization of some metric defined upon the corresponding model.

[Choi and Lee, 2011] propose a mixed integer linear program to minimize the reader interference problem as well as other performance metrics by selecting channel, time slots and output power for each reader. Their strategy is based on achieving a minimal Signal to Interference plus Noise Ratio (SINR) for the signal received from tags, as well as on maximizing network utilization and minimizing power consumption. However, they neglect the availability of information about the number of tags present in the reading area of each reader, as well as the operation of the underlying reading protocols, which are major factors determining the performance of the reading process.

[Kim et al., 2007] propose the TPC-CA algorithm based on a power control approach. It consists of controlling the reader output power optimally to reduce reader-to-reader collisions. Optimality criterion is related to minimize the area where interferences among readers occurs.

Chiu et al. introduces GA-BPSO in [Chui-Yu et al., 2009] a scheduler based on Genetic Algorithm and Swarm Intelligence meta-heuristics for single-channel environments. These schedulers aims at minimizing overall sum of transaction times. However, these times are provided as parameters for the scheduler, and are not based on the impact of resources allocation on the reading protocols.

[Deolalikar et al., 2005] derive in optimal scheduling schemes for readers in RFID networks for four basic configurations. As in our work authors aim at maximizing the number of identification within the scheduling period (t). But they model the performance of the reading process with an approximation: the number of tags identified increases linearly up to a saturation point. From that point on, the number of identifications remains constant. As we demonstrate in Section 1.2.1 this approach is not realistic for different tag-to-tag anti-collision protocols configurations (e.g. in Static-FSA there is a drop on the throughput). As in our work only reader-to-reader interferences (and thus single-channel) environments are considered.

[Mohsenian-Rad et al., 2010] is the work more closely related to ours. Authors design two optimization-based distributed channel selection and randomized interrogation algorithms for dense RFID systems: FDFA (which is fully distributed and achieves a local optimum) and SDFA (semi-distributed and reach to the global optimum). In addition, authors realistically assume that reader may operate asynchronously. Similarly to our work, they consider a FDMA/TDMA scheduler, where the medium access control layer of the readers complies with EPCglobal Class-1 Gen-2 standard (and therefore a reader may allocate a number of interrogation frames within its allocated time). In this work authors focus on the probability that a reader starts an interrogation interval without experiencing either reader-to-reader or reader-to-tag collisions. The goal is to achieve max-min fairness in the network, and as a result the processing load is evenly distributed among all readers. However, this Chapter does not consider the knowledge about the number of contending tags in range of each reader. This information allow us to formulate the optimization problem in terms of reading efficiency (maximizing the number of tags identified in the overall time period). An additional contribution of [Mohsenian-Rad et al., 2010] is to develop a protocol to construct the topology (*i.e.* reader-to-reader and reader-to-tag constraints) by exchanging some messages in three control channels. This protocol may be implemented in other schedulers (like ours) to determine the network topology in real-time.

[Tanaka and Sasase, 2007] also determine an interference model which they apply later to formulate constraints in a binary integer linear program aimed at maximizing the ratio of total time where readers can successfully communicate with the tags and total interrogation time of readers. As in our model, the goal is selecting suitable time slot and channels for each reader. They also propose two heuristics (one distributed and one centralized) for solving the allocation problem efficiently.

[Seo and Lee, 2010] propose a FDM/TDM scheduler (RA-GA) based on a reader-to-reader interference model, which seeks to maximize an utility function depending upon the operating time slots. This problem is solved by using a genetic algorithm meta-heuristic.

As many of the previous works, neither in [Tanaka and Sasase, 2007] nor in [Seo and Lee, 2010] the reading protocol or the current load (unidentified tags) of each cell are considered.

Summarizing, to the best of our knowledge, all previous optimization models ignore the availability of information about the number of tags within range of each reader. This information can be very effectively exploited to construct a scheduler with the goal of maximizing the number of identifications in the interrogation period. Besides, most previous works assume a model view from the physical layer perspective, and are usually aimed at minimizing interference. This view has notable limitations since tag identification performance, and thus scheduling, heavily depends on the underlying tag-to-tag anti-collision protocol, as discussed in next Section.

1.2.1. Tag identification procedure

The identification process involves communications between the reader and the tags and takes place in a shared wireless channel. Basically, the reader *interrogates* tags nearby by sending a *Query* packet (the exact format of this packet depends on the particular standard). Tags are energized by the reader's signal and respond to this request with their identification. When several tags answer simultaneously, a tag-to-tag collision occurs, and the information cannot be retrieved. Therefore, a tag-to-tag anti-collision mechanism is required when multiple tags are in range. Aloha-based protocols, also called probabilistic or random access protocols, are the most prevalent in the UHF band. They are designed for situations in which the reader does not know exactly how many tags will cross its checking area. The most common Aloha RFID protocol is FSA, a variation of Slotted-Aloha. As in Slotted-Aloha, time is divided into time units called slots. However, in FSA, slots are subject to a super-structure called a "frame". Two options of FSA are commonly used in RFID technology:

1. **Static frame length FSA (Static-FSA).** The reader starts the identification process with an identification frame by sending a *Query* packet with information about the frame length (k slots) to the tags. The frame length is kept unchanged during the whole identification process. At each frame, each unidentified tag selects a slot at random from among the k slots to send its identifier to the reader. FSA achieves reasonably good performance at the cost of requiring a central node (the reader) to manage slot and frame synchronization. FSA has been implemented in many commercial products and has been standardized in the ISO/IEC 18000-6C [ISO, 2010a], ISO/IEC 18000-7 [ISO, 2010b] and EPCGlobal Class-1 Gen-2 (EPC-C1G2) standards [EPCglobal, 2013].

2. **Dynamic frame length FSA (Dynamic-FSA).** When tags outnumber available slots, identification time increases considerably due to frequent tag-to-tag collisions. On the other hand, if the slots outnumber the tags, many slots will be empty in the frame, which also leads to long identification times. Dynamic-FSA protocols were conceived to address this problem. They are similar to Static-FSA but the number of slots per frame is variable. In other words, parameter k may change from frame to frame in the *Query* packet to adjust the frame length. Dynamic-FSA operation is optimal in terms of reading throughput (rate of identified tags per slot) when the frame length equals the number of contenders [Abramson, 1973]. Therefore, to maximize throughput the reader should ideally know the actual number of competing tags and allocate that number of slots to the next frame. Different Dynamic-FSA algorithms have been proposed to estimate the number of competing nodes based on the collected statistical information. The most relevant ones have been studied in depth in previous papers [Bueno-Delgado et al., 2009; Vales-Alonso et al., 2011].

In the next Section, both algorithms (Static-FSA and Dynamic-FSA) are considered in order to propose an optimal slot distribution for the single channel environment. In the case of Static-FSA, the frame length is k for all readers, and in the case of Dynamic-FSA we are assuming that each reader j actually knows the number of competing nodes at frame i ($n_{j,i}$), and that the reader is adjusting $k_{j,i} = n_{j,i}$ if the number of the remainder available slots is greater than $n_{j,i}$.

1.3. Optimal time distribution

Recall from the introduction that it is assumed a dense-reader environment with the limitation of a single frequency channel f=1, m readers, and t slots available in the channel. In addition, for each reader j=1,...,m, let us denote:

- n_j, the tags unidentified in the range of the reader j
- t_j, the number of slots assigned to reader j

Let us remark that the methods used in Dynamic-FSA tag-to-tag anti-collision protocols to determine the number of contenders can be directly applied in our case to estimate n_j in real-time (see [Bueno-Delgado et al., 2009] and [Vales-Alonso et al., 2011] for details).

Besides, topological dependencies among readers are defined by a $m \times m$ matrix $A = (a_{jj'})$ whose elements are 1 if reader j and j' cannot operate at the same time, and 0 otherwise.

Let $\varphi(n,t)$ denote the expected number of identified tags when n tags contend in t slots, and let us define Φ as the whole expected number of identified tags in the network, that is,

$$\Phi = \sum_{j=1}^{m} \varphi(n_j, t_j) \tag{1.1}$$

Then, the optimization problem can be stated as solving:

$$\max_{\substack{t_j \\ j=1,\ldots,m}} \Phi \tag{1.2}$$

Subject to

$$t_j \geq 0 \tag{1.3}$$

and

$$t_j + I_{t_j} \sum_{j'=1, j' \neq j}^{m} t_{j'} a_{j'j} \leq t, \text{ for all } j=1,\ldots,m \tag{1.4}$$

where I_{t_j} is 1 if t_j is greater than 0, and 0 otherwise.

Constraint (1.3) expresses a basic limiting condition on the values assigned to the number of assigned slots.

The key in our problem formulation is constraint (1.4) which establishes local conditions to regulate the spatial reuse of resources in our network. This condition states that the number of slots assigned to a reader j plus those assigned to its neighbours can not surpass the number of available slots. I_{t_j} is included since readers without slots assigned should be considered as disconnected, and no constraints have to be applied to that particular readers.

The former constraint guarantees that enough slots are available for each node in each neighbourhood (set of nodes bonded with topological constraints, *i.e.* $a_{j'j}$=1) to obey with the limit of t slots among all neighbours. Note that it does not guarantee that these slots can be allocated consecutively. However, this is not an issue since tags do not proceed with the next slot until a *QueryRep* packet arrives from the reader. Hence, even if slots are not consecutively allocated, tags perceive continuity and the identification can be performed seamlessly.

We must remark that this set of constraints produces feasible solutions regardless of the considered topology. However, in some cases (as we will show in next Section) the constraint is too strict and may lead to suboptimal solutions since space reutilization is limited. If the network graph has a large density (i.e. the number of edges is close to the maximal number of edges) the results provided by solving problem (1.1) will be close to the optimal solution with maximal space resource reutilization. Whereas, for sparse network graphs the space reutilization will be small. The first kind of scenario will likely occur (due to the large reader-to-reader interference range) in facilities with non-screened readers, and thus the

solutions found will be realistic.

Finally, in order to solve the optimization problem the expected number of identifications $\varphi(n,t)$ must be computed. Next Sections deal with its computation both for Static-FSA and for Dynamic-FSA.

1.3.1. $\varphi(n_j,t_j)$ computation for Static-FSA

In this case, the reading process for each reader j consists of several consecutive reading frames of length k, until all the t_j reading slots are eventually exhausted. It is assumed that $t_j{=}ka$, being a a positive integer or zero. Given the last condition, and since expectation is a linear operator, $\varphi(n_j,t_j)$ can be computed as the sum of the average number of tags identified in the first frame ($\varphi(n_j,k)$) plus those identified in the remainder process ($\varphi(n_j-\eta,t_j-k)$), where η denotes the random number of tags identified in the first frame.

The former part can be computed if the distribution of the random variable η is known, so let us denote $P(a|n_j,k)$ the probability that a tags are identified if n_j tags contend in k slots. Then

$$\varphi(n_j,k)=\sum_{a=0}^{n_j} aP(a|n_j,k) \tag{1.5}$$

The second part, can be computed given the joint probability of identifying a tags in the first frame and a' in the remainder process if n_j tags contend in t_j slots, which we will denote as $P(a,a'|n_j,t_j)$. We obtain:

$$\varphi(n_j-\eta,t_j-k)=\sum_{a=0}^{n_j}\sum_{a'=0}^{n_j} a'P(a,a'|n_j,t_j) \tag{1.6}$$

But, clearly $P(a,a'|n_j,t_j)=P(a|n_j,k)P(a'|n_j-a,t_j-k)$ and this leads to

$$\varphi(n_j-\eta,t_j-k)=\sum_{a=0}^{n_j}\varphi(n_j-a,t_j-k)P(a|n_j,k) \tag{1.7}$$

Appendix A demonstrates that the value of $P(a|n_j,t_j)$ is given by:

$$P(a|n_j,t_j)=\frac{n_j!}{t_j^{n_j}}\binom{t_j}{a}\sum_{c=0}^{n_j-a}(-1)^c\binom{t-a}{c}\frac{(t_j-a-c)^{n_j-a-c}}{(n_j-a-c)!} \tag{1.8}$$

Thus, the following recursive equation results:

$$\varphi(n_j,t_j)=\begin{cases}\sum_{a=0}^{n_j}(a+\varphi(n_j-a,t_j-k))P(a|n_j,k), \text{ if } t_j\geq k,\\ 0, \text{ otherwise}\end{cases} \tag{1.9}$$

1.3.2. $\varphi(n_j, t_j)$ computation for Dynamic-FSA

In this second case, the reading process for each reader j also consists of several reading frames but of variable length $k_{j,1}, k_{j,2}, \ldots$, until all the t_j reading slots are exhausted. Besides, denote the number of contenders in each frame is $n_{j,1}, n_{j,2}, \ldots$. Since Dynamic-FSA operation is used (see Section 1.2.1), the reader seeks to maximize reading throughput and allocates the optimal number of slots in each frame. That is, as much slots as the number of contending tags ($k_{j,i}$=$n_{j,i}$). This is possible while $n_{j,i} < t_j - \sum_{c=1}^{i-1} k_{j,c}$, that is, if the remainder number of slots is greater that the number of contenders. Otherwise we assume that a last frame is allocated with all the remaining slots ($k_{j,i} = t_j - \sum_{c=1}^{i-1} k_{j,i}$).

Like in the previous case $\varphi(n_j, t_j)$ can be described through a recursive equation,

$$\varphi(n_j, t_j) = \begin{cases} \varphi(n_j, n_j) + \sum_{a=0}^{n_j} \varphi(n_j - a, t_j - n_j) P(a|n_j, n_j) \text{ if } n_j < t_j \\ \varphi(n_j, t_j) \text{ if } n_j \geq t_j \end{cases}$$

from Eq. (1.5),

$$\varphi(n_j, n_j) = \sum_{a=0}^{n_j} a P(a|n_j, n_j)$$

and,

$$\varphi(n_j, t_j) = \sum_{a=0}^{n_j} a P(a|n_j, t_j), \quad \text{if } n_j \geq t_j$$

Hence,

$$\varphi(n_j, t_j) = \begin{cases} \sum_{a=0}^{n_j} (a + \varphi(n_j - a, t_j - n_j)) P(a|n_j, n_j) \text{ if } n_j < t_j \\ \sum_{a=0}^{n_j} a P(a|n_j, t_j) \text{ if } n_j \geq t_j \end{cases} \tag{1.10}$$

1.4. Results

The optimal assignment has been computed in Static and Dynamic-FSA cases using the recursive formulas described in the previous Section. Three representative scenarios (see Figure 1.1) have been selected. Edges (i.e. connecting lines) represent the existence of interference between two vertices (readers). On the first scenario, a full-mesh topology of m readers has been selected. It is a typical configuration in facilities, since the RRI distance is large (in the order of hundreds of meters) as discussed in the introduction. On the other hand, the star topology of m readers selected for scenario 2 represents another practical case, where readers are confined to some areas (*e.g.* by screening the reading area), and interferences are restricted to some particular pairs, exclusively between R1 and the other readers in this example. Finally, the line scenario is representative of an assembly line, where

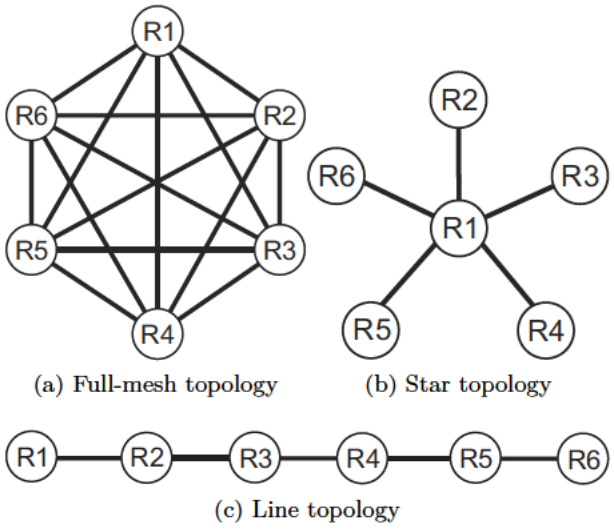


Figure 1.1: Example scenarios for $m = 6$

neighbor readers are in range.

Besides, the following parameters have been considered:

- $t = 512$,
- n tags to be identified at each reader, from 1 to 100 tags,
- $m = 2$, 4, 6, 8 and 10,
- and for Static-FSA $k = 16$ and 64.

Our optimization algorithm has been implemented using the *General Algebraic Modeling System* (GAMS), a high-level modeling system for mathematical programming and optimization, and AlphaECP, a MINLP (Mixed-Integer Non-Linear Programming) solver based on the extended cutting plane (ECP) method. It allowed us to define our optimization problem directly from the mathematical description provided in Section 1.3.

Tables 1.1, 1.2, and 1.3 show the optimal configurations (slots assigned to each reader) for the Dynamic-FSA protocol in all scenarios with $m = 4$. Let us remark that the optimal solutions are non-trivial, that is, can be obtained through an educated guess. Clearly, this solution is not unique: a circular permutation of the optimal solution, replacing the slots from Rj to R$(j+1)$ if $j < m$ and from Rm to R1 is also an optimal solution for the full-mesh scenario. The same applies to the star scenario if the slots in R1 are kept constant while any permutation is applied to the rest of the readers, and to the line scenario replacing the slots from Rj to R$(m-j)$.

n	Φ	R1	R2	R3	R4
10	40.000	128	128	128	128
20	80.000	126	115	155	116
30	119.999	128	128	128	128
40	159.427	128	128	128	128
50	186.355	128	128	128	128
60	189.195	94	140	139	139
70	189.113	110	70	166	166
80	188.949	80	162	190	80
90	188.992	0	212	89	211
100	188.905	157	157	99	99

Table 1.1: Full-mesh scenario. Optimal assignment of slots for Dynamic-FSA

n	Φ	R1	R2	R3	R4
10	40.000	78	74	76	77
20	80.000	126	115	155	116
30	119.999	128	128	128	128
40	159.427	128	128	128	128
50	186.355	128	128	128	128
60	189.195	94	140	139	139
70	210.000	0	512	512	512
80	240.000	0	512	512	512
90	270.000	0	512	512	512
100	300.000	0	512	512	512

Table 1.2: Star scenario. Optimal assignment of slots for Dynamic-FSA

n	Φ	R1	R2	R3	R4
10	40.000	85	86	165	221
20	80.000	118	118	124	248
30	120.000	153	153	171	171
40	160.000	173	170	169	173
50	199.913	174	169	169	174
60	236.153	176	168	168	176
70	254.481	188	162	162	188
80	263.693	215	172	125	215
90	273.451	242	180	90	242
100	287.462	256	256	0	453

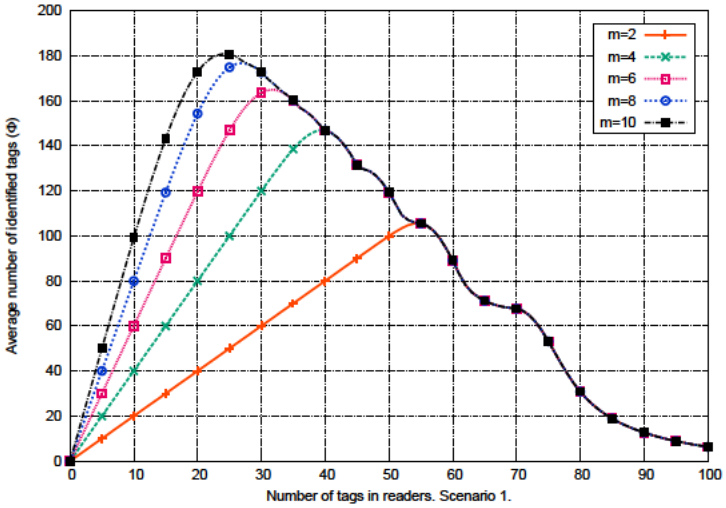
Table 1.3: Line scenario. Optimal assignment of slots for Dynamic-FSA

In addition, note that results obtained for the Star scenario (Table 1.2) can be improved. For example, for $n = 30$, after assigning 128 slots to R1 it would be possible to assign 384 to all remainder readers, which will provide a solution better than that obtained by solving problem (1.1). As discussed in Section 1.3 this is caused by the strict resource reutilization obtained by applying the set of constraints given by Eq. (1.4). This problem does not appear for networks characterized by a dense graph, as the Full-mesh scenario.

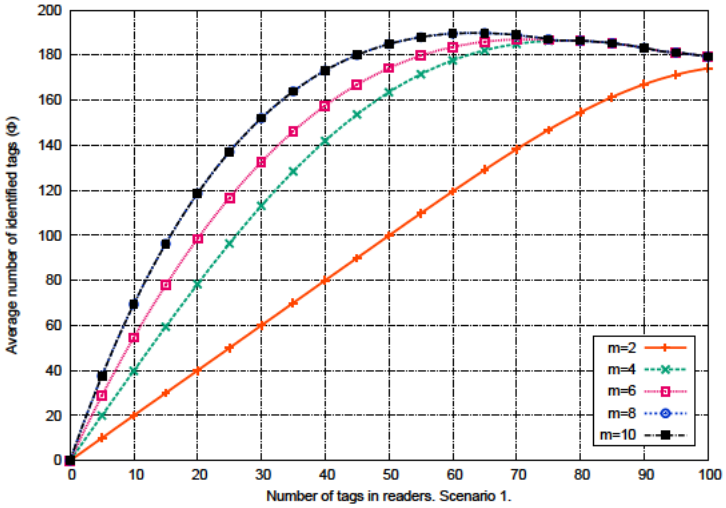
Besides, Figures 1.2, 1.3, and 1.4 show the expected number of tags identified (Φ) for all the possible values of m using the optimal assignments. Note that the resources available (t=512) are the same for all the configurations, however the performance clearly varies. This illustrates how the underlying reading protocol determines the final system performance. Dynamic-FSA performs better than Static-FSA assignment for both configurations of k (16, 64), as can be expected. This is reasonable since Dynamic-FSA achieves an optimal reading throughput frame-by-frame while the number of available slots is at least equal to the number of contenders.

Another important result shown in these figures, is the existence of saturation points in the system. That is, in some cases the throughput does not increase when the load is increased. For Dynamic-FSA, in all cases, the throughput never decreases, this is caused by the flexibility of Dynamic-FSA to adapt to different loads. For Static-FSA k=64, the effect is almost similar to the Dynamic-FSA case, except in the Full-mesh scenario where beyond n=60 tags the throughput slightly decreases. However, for all Static-FSA k=16 cases the effect of the load in the throughput is dramatic, with a throughput minima and a step decreasing performance. This is of considerable importance, since Static-FSA k=16 is the default configuration of many readers in the market, and this configuration leads to poor collective performance.

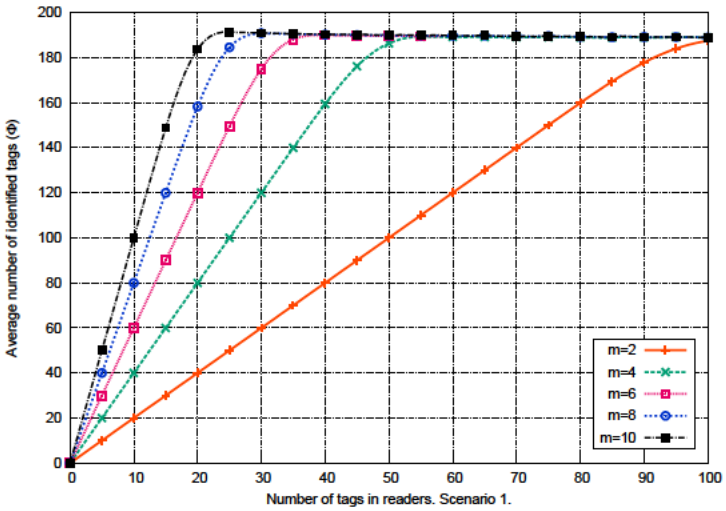
In addition Figures 1.5, 1.6, and 1.7 show, for m=4, the performance of the optimal allocation versus a non-optimal allocation scheme selected for comparison. Namely using $\frac{1}{m}$ of time allocated to each reader ("proportional" resource sharing), that is, t_1=...=t_4=128. This heuristic is a natural choice, since the number of tags in range of each reader is the same, therefore a good performance could be expected. In fact, the proportional scheme achieves in a range of n a performance nearly equal to the optimal one, as can intuitively be expected, in some cases (e.g. Static-FSA k=64 and Dynamic-FSA in the Full-mesh topology). However for some cases the allocation is clearly suboptimal (e.g. Star and Line scenarios for n>60). Noteworthy, in the star scenario, there is a point ($n \geq 70$) where the best option is directly to disconnect the central reader. In this case, without restrictions in the network, the remainder readers can be allocated each all the 512 slots. A similar behaviour occurs for the Line scenario disconnecting some readers when there is a high load (e.g. see Table 1.3, if n=100 the third reader is disconnected).



(a) Static-FSA k=16

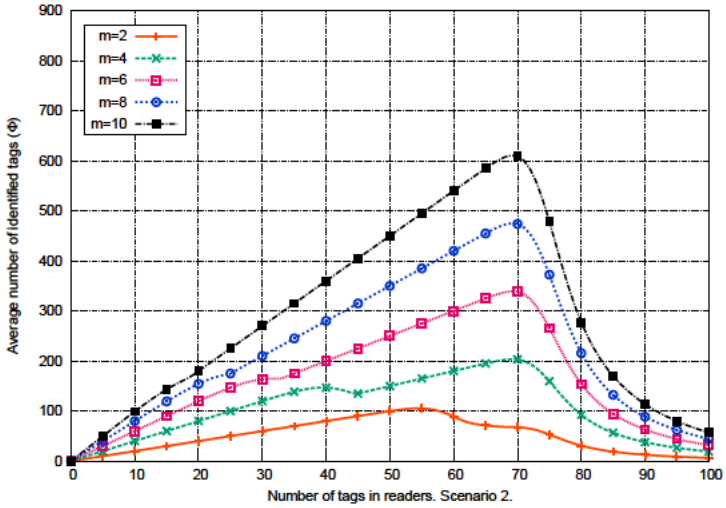


(b) Static-FSA k=64

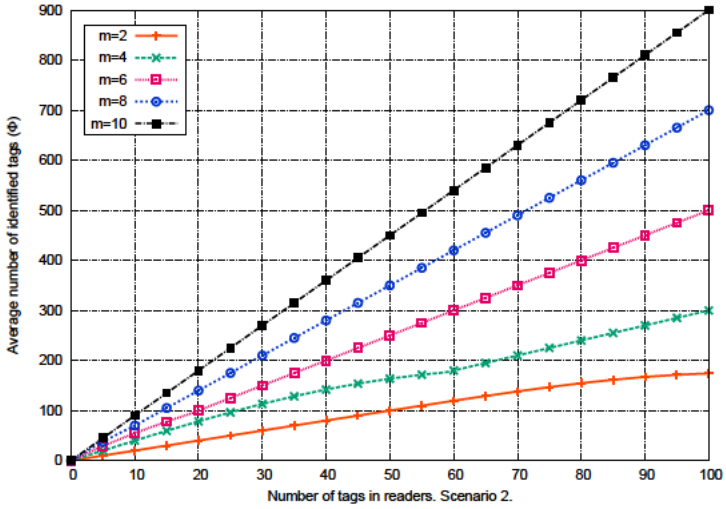


(c) Dynamic-FSA

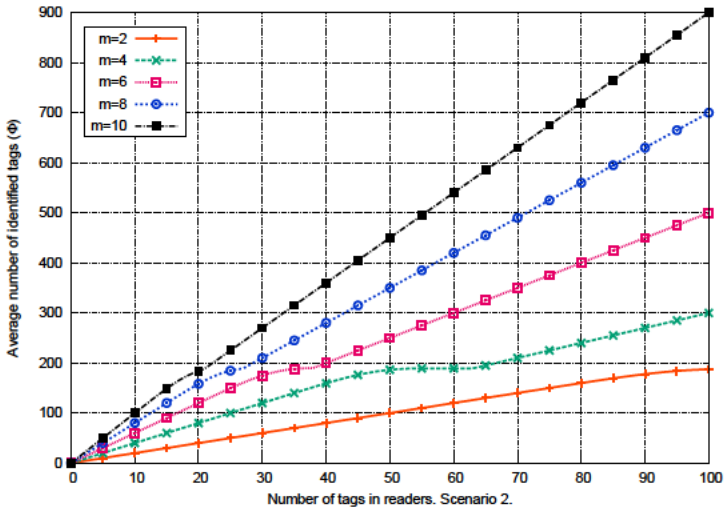
Figure 1.2: Expected number of identifications (Φ) versus n. Full-mesh scenario



(a) Static-FSA k=16

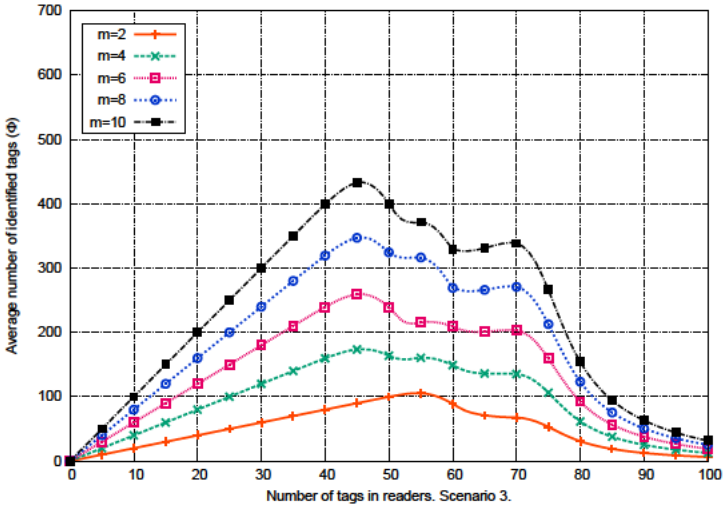


(b) Static-FSA k=64

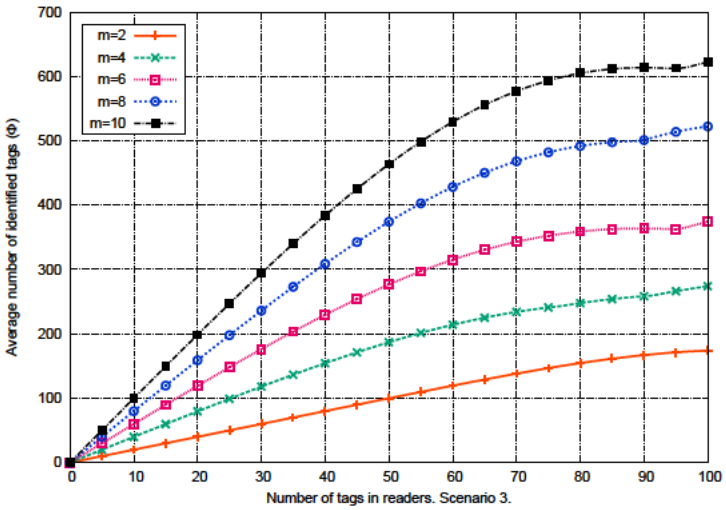


(c) Dynamic-FSA

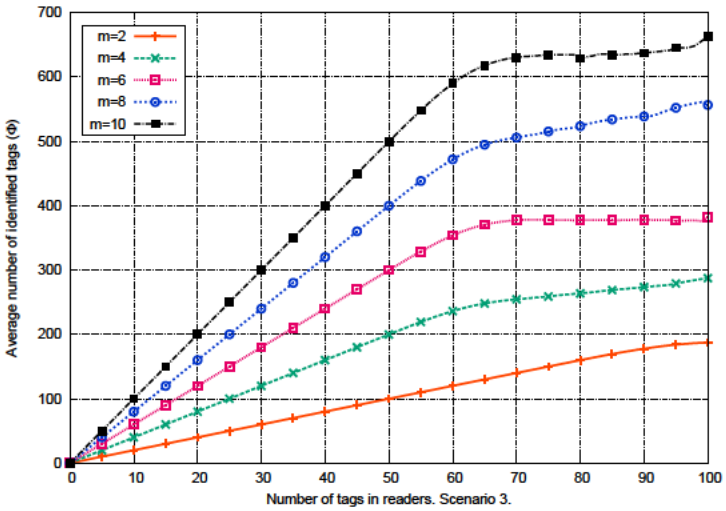
Figure 1.3: Expected number of identifications (Φ) versus n. Star scenario



(a) Static-FSA k=16

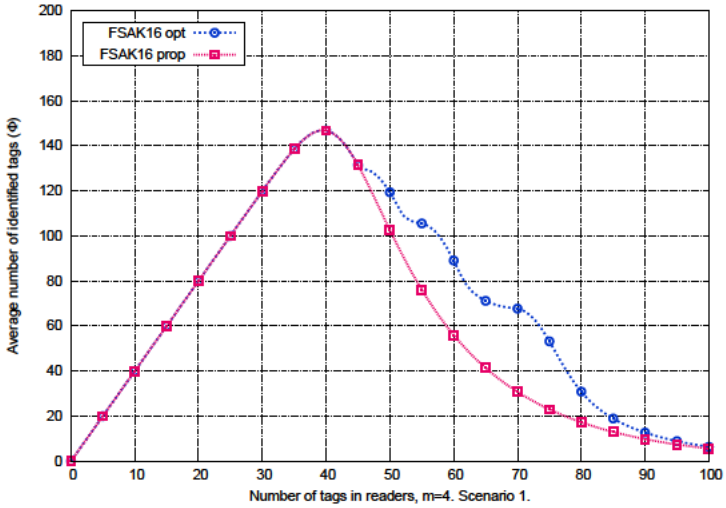


(b) Static-FSA k=64

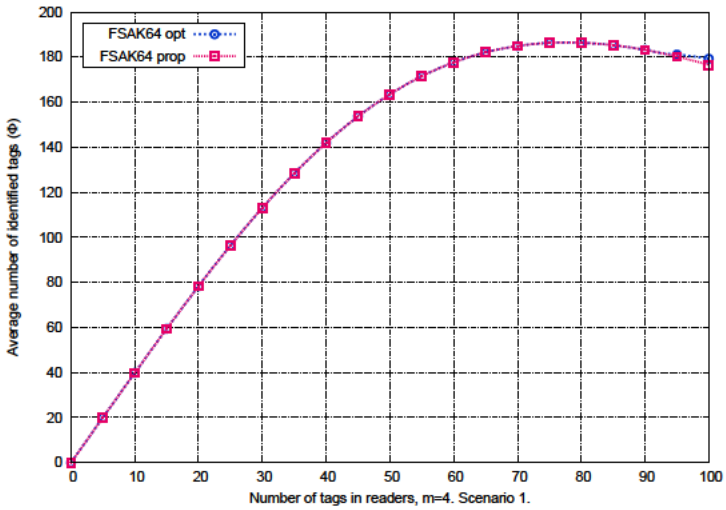


(c) Dynamic-FSA

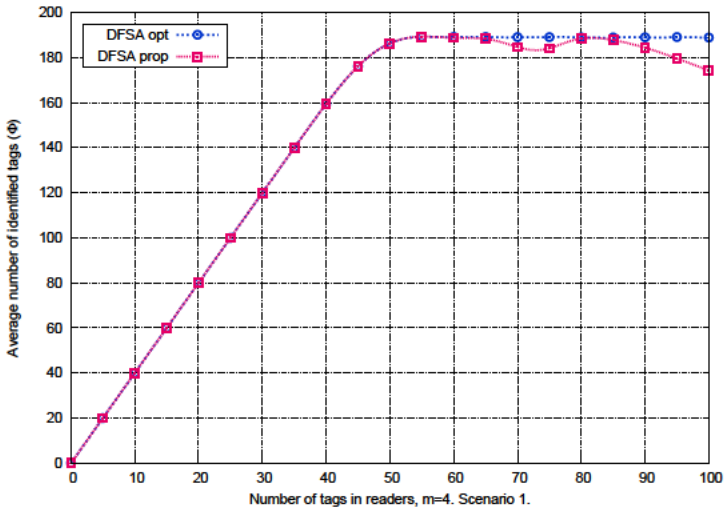
Figure 1.4: Expected number of identifications (Φ) versus n. Line scenario



(a) Static-FSA k=16

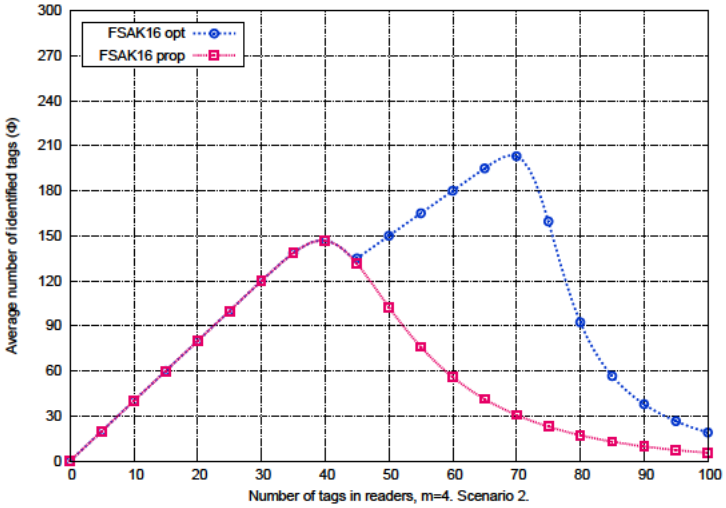


(b) Static-FSA k=64

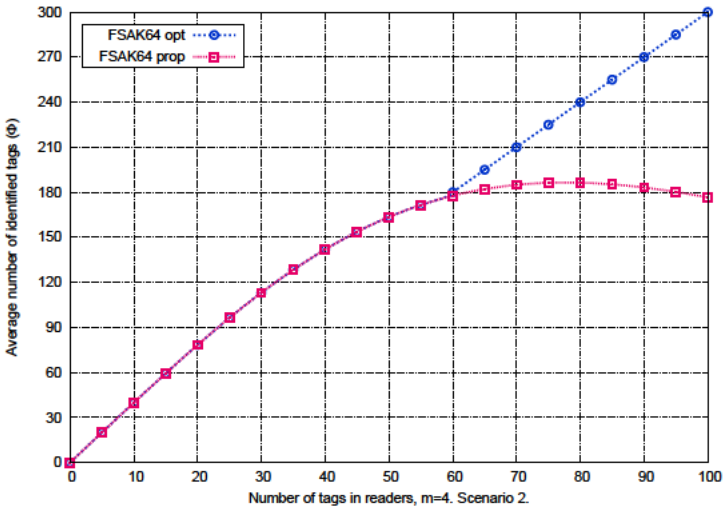


(c) Dynamic-FSA

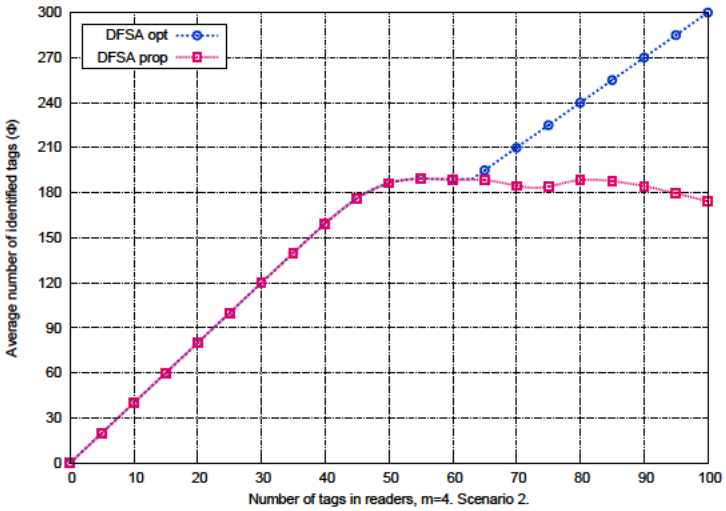
Figure 1.5: Optimal vs proportional allocation for Full-mesh topology, $m = 4$



(a) Static-FSA k=16

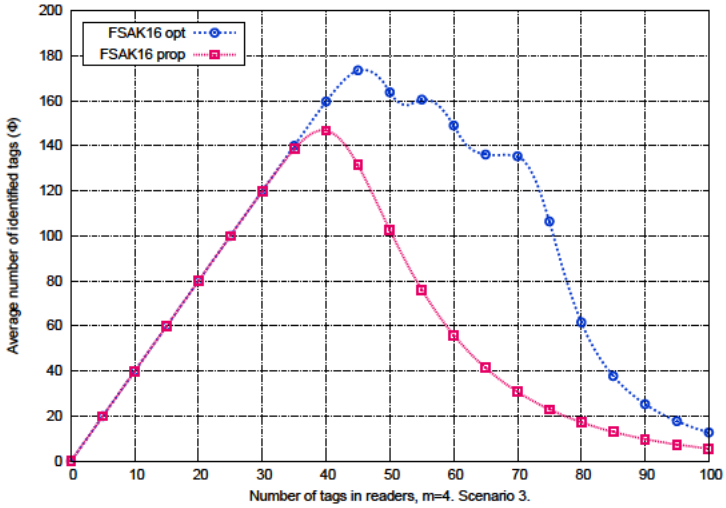


(b) Static-FSA k=64

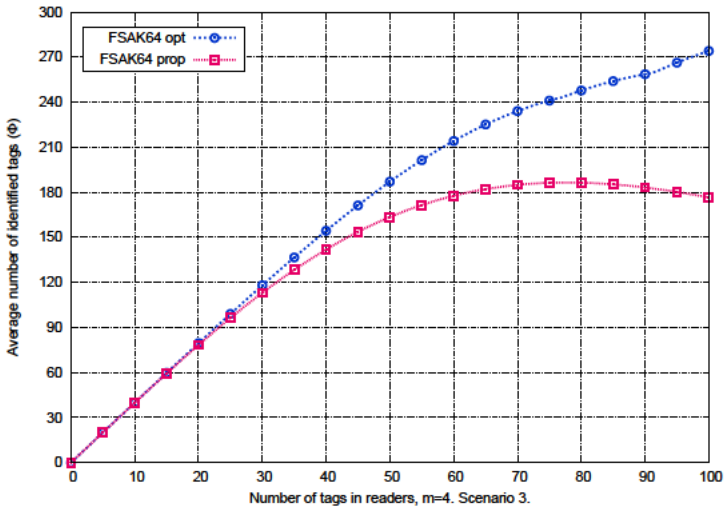


(c) Dynamic-FSA

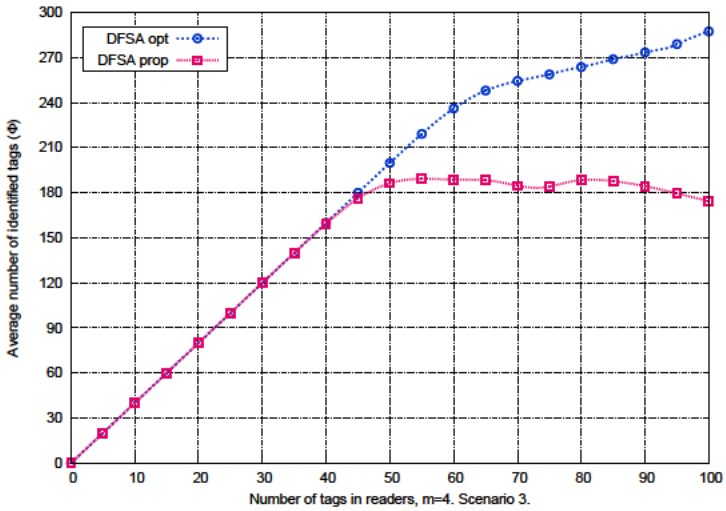
Figure 1.6: Optimal vs proportional allocation for Star topology, $m = 4$



(a) Static-FSA k=16



(b) Static-FSA k=64



(c) Dynamic-FSA

Figure 1.7: Optimal vs proportional allocation for Line topology, $m = 4$

1.5. Tag estimation impact on scheduler performance

The aim at this Section is twofold:

1. Quantify the improvement achieved in the scheduler when tag instant population estimation is available.

2. Quantify the impact of tag population estimation errors on the performance achieved by the scheduler.

As stated in Section 1.2 previous works do not assume knowledge about the tag population, and are mostly based on minimizing interferences. To establish a comparison between our model and a reference model that do not use population information at least we must focus on the same performance metric, *i.e.* the expected number of identifications (which can also be viewed as throughput).

Although our reference model does not use information about the instant population is rational to assume at least a coarse knowledge of the environment, typically the average number of competing tags. This allows the designer to configure the system for a standard case. Note that if this information is unavailable the designer should guess somehow a configuration, and the performance would be lower than in the reference model.

Henceforth, let us assume that our reference model is based on the availability of information about the average tag population, and that the designer is able to select the optimal scheduler configuration for this case (*e.g.* by solving problem (1.1)).

For simplicity, let us denote by $\vec{n}$ the m-dimensional vector $(n_1, \ldots, n_m)$, and by $\Phi_{\vec{n}}(\vec{n'})$ the expected number of identifications when the optimal solution to problem (1.1) with tag estimation parameter $\vec{n}$ is applied to the actual population $\vec{n'}$. Besides, let $\vec{n^*}$ denote the m-dimensional vector where the jth component is the average number of tags in reader j.

Thus, if the probability distribution of tag population, *i.e.* $P(\vec{n})$ is known, the improvement (Δ) achieved by our scheduler over the reference scheduler is:

$$\Delta = \sum_{\forall \vec{n}} (\Phi_{\vec{n}}(\vec{n}) - \Phi_{\vec{n^*}}(\vec{n}))P(\vec{n}) \tag{1.11}$$

By solving problem (1.1) both optimal slot assignments can be computed. Let us denote $\widehat{t}$ and $\widehat{t^*}$ to the optimal assignments for tag population $\vec{n}$ and $\vec{n^*}$ respectively, and $\widehat{t}_j$ and $\widehat{t^*}_j$ the slots assigned to particular reader jth.

Then, Δ can be rewritten as:

$$\Delta = \sum_{\forall \vec{n}} (\sum_{j=1}^{m} \varphi(n_j, \widehat{t}_j) - \sum_{j=1}^{m} \varphi(n_j, \widehat{t^*}_j))P(\vec{n}) \tag{1.12}$$

m	Dynamic-FSA	Static-FSA, K=64	Static-FSA, K=16
2	0.0273	0	0.0889
4	0.0120	0.0008	0.2891
6	0.0423	0.0050	0.4831
8	0.0791	0.0086	0.6024
10	0.0947	0.0086	0.6731

Table 1.4: Full-mesh scenario. Ratio of improvement using tag estimation. $\epsilon = 0$

where $\varphi(n,t)$ is computed directly with formulas (1.9) and (1.10) for Static-FSA and Dynamic-FSA, respectively.

In addition, it could be argued that instant tag population estimation may be subject to errors. This can be included in our computations through an error vector $\vec{\epsilon}$, where the jth component is $\epsilon_j = \breve{n_j} - n_j$, n_j the estimation, and $\breve{n_j}$ is the actual number of tags. Therefore, the real tag distribution is $\vec{n} + \vec{\epsilon}$, and Δ should be modified as:

$$\Delta = E_{\vec{\epsilon}}\{\sum_{\forall \vec{n}}(\sum_{j=1}^{m}\varphi(n_j+\epsilon_j,\widehat{t_j}) - \sum_{j=1}^{m}\varphi(n_j+\epsilon_j,\widehat{t^*}_j))P(\vec{n})\} \tag{1.13}$$

Note that in this last case, $\widehat{t_j}$ is still computed with the estimation vector $\vec{n}$. So, if $\vec{n}$ and $\vec{\epsilon}$ are independent we finally reach to:

$$\Delta = \sum_{\forall \vec{\epsilon}}[\sum_{\forall \vec{n}}(\sum_{j=1}^{m}\varphi(n_j+\epsilon_j,\widehat{t_j}) - \sum_{j=1}^{m}\varphi(n_j+\epsilon_j,\widehat{t^*}_j))P(\vec{n})]P(\vec{\epsilon}) \tag{1.14}$$

Note that we use a perfect knowledge of the average number of tags, therefore we are assuming the least-favorable comparison case for our scheduler versus the reference model.

1.5.1. Numerical examples

For the sake of example, let us assume that, for each reader, the number of tags is given by a uniformly distributed random variable $\mathbf{n}$ in the range $[0, 100]$. That is, $n_j = \mathbf{n}$ for all $j = 1, \ldots, m$. Hence, $\vec{n^*} = (50, \ldots, 50)$. Tables 1.4, 1.5, and 1.6 show the average performance improvement achieved for the examples described in the previous Section and for the Dynamic-FSA and Static-FSA tag-to-tag anti-collision protocols. Results are shown as the ratio of improvement (Δ) to the expected readings without using tag estimation.

Results clearly depends on the scenario and on the tag-to-tag anti-collision protocols. Improvement ranges from nearly 0% in many Static-FSA k=64 cases, while it may reach up to 67% for Static-FSA, k=16 in the full-mesh scenario. For Dynamic-FSA improvement is between 2,7% and 21,14%, depending on the particular scenario. Let us remark again that this comparison is performed against

m	Dynamic-FSA	Static-FSA, K=64	Static-FSA, K=16
2	0.0273	0	0.0889
4	0.1592	0.1702	0.2636
6	0.0509	0.0023	0.1940
8	0.0417	0.0005	0.1756
10	0.0333	0.0000	0.1683

Table 1.5: Star scenario. Ratio of improvement using tag estimation. $\epsilon = 0$

m	Dynamic-FSA	Static-FSA, K=64	Static-FSA, K=16
2	0.0266	0	0.0817
4	0.0331	0.0144	0.2114
6	0.0049	0.0018	0.2105
8	0.0298	0.0050	0.2073
10	0.0101	0.0013	0.2056

Table 1.6: Line scenario. Ratio of improvement using tag estimation. $\epsilon = 0$

the average tags identified when the optimal configuration is computed using as information the mean number of competing tags. Therefore, this is the *minimum* improvement ratio: non-optimal schedulers (as the reference heuristic used in Section 1.4) will obtain worse results.

Besides, we can consider the estimation error. In our test we have assumed for each reader an error distributed uniformly $\epsilon \sim U[-10, 10]$. Results are shown in Tables 1.7, 1.8, and 1.9.

Again, the results heavily depend on the configuration, but in most cases even assuming an error in the tag number estimation, they show a positive feedback by using the estimation. In some cases in the Full-mesh and Line scenarios there is a negative impact, but almost negligible. Therefore, we can conclude that even assuming errors, the utilization of tag estimators are worth to be considered.

1.6. Conclusions

This Chapter introduced a novel optimal scheduler for a particular dense reader environment composed by m readers which must share a single frequency channel. The scheduler proposed exceeds in performance to heuristic algorithms, improving the average number of tags identified in an RFID facility. Besides, the effect of the reading protocols has also been studied in depth, concluding that a dynamic FSA algorithm excels static frame length ones. Indeed, the impact of using knowledge about tag population in the scheduler has been analyzed. It has been concluded that, even assuming errors in the estimation, our scheduler is able to obtain a higher performance than a reference model where the average population is perfectly known.

As future works we aim at extending our model to multi-channel scenarios, developing a model that allow full resource reutilization, and further analyze RFID realistic scenarios to propose optimal configuration strategies.

m	Dynamic-FSA	Static-FSA, k=64	Static-FSA, k=16
2	0.0284	0	0.0527
4	0.0064	0.0007	0.2334
6	0.0361	0.0040	0.4107
8	0.0580	0.0101	0.5515
10	0.0747	0.0070	0.5712

Table 1.7: Full-mesh scenario. Ratio of improvement using tag estimation. $\epsilon \sim U[-10, 10]$

m	Dynamic-FSA	Static-FSA, k=64	Static-FSA, k=16
2	0.0275	0	0.0704
4	0.1541	0.1750	0.2087
6	0.0543	0.0017	0.2123
8	0.0406	-0.0002	0.1878
10	0.0302	0.0000	0.1925

Table 1.8: Star scenario. Ratio of improvement using tag estimation. $\epsilon \sim U[-10, 10]$

m	Dynamic-FSA	Static-FSA, k=64	Static-FSA, k=16
2	0.0442	-0.0006	0.0023
4	0.0324	0.0212	0.0991
6	0.0003	0.0019	0.1169
8	0.0279	0.0071	0.0815
10	0.0118	0.0012	0.0942

Table 1.9: Line scenario. Ratio of improvement using tag estimation. $\epsilon \sim U[-10, 10]$

Appendix A. Computation of $P(a|n,t)$

To compute the probability $P(a|n,t)$, we apply the technique in [Milenkovic and Compton, 2004], where the authors formulate probabilistic transforms for urn models that convert the dependent random variables describing urn occupancies (slot occupancies in our case) into independent random variables. Due to the independence of random variables in the transform domain it is simpler to compute the statistics of interest, and afterwards the transform is inverted to get the desired result.

Let us denote $P(a|n,t)$ as the probability of interest and $\boldsymbol{P}(\lambda,t,i)$ its transformation, being λ a parameter meaningful in the transform domain only. Indeed, there is no dependence on the number of balls (tags), n, in the transform domain.

The procedure is as follows: first, the appropriate transform for a particular urn model is selected. In our case, both the t urns (slots) and the n balls (tags) are distinguishable. In this case, the independent random variables $Z_1,\ldots,Z_t$ describing the occupancy of an urn in the transform domain are Poisson distributed with mean λ [Milenkovic and Compton, 2004]. That is, $P(Z_i=j)=e^{-\lambda}\frac{\lambda^(j)}{j!}$. Second, the probability of interest, $\boldsymbol{P}(\lambda,t)$, is computed in the transformed domain. In our case, given a frame of length t, and taking into account the independence of Z_i, the probability of having i urns (slots) with one ball (tag) is:

$$\begin{aligned}\boldsymbol{P}(\lambda,t) &= \binom{t}{i} P(Z=1)^i (1-P(Z=1))^{t-i} \\ &= \binom{t}{i} (e^{-\lambda}\lambda)^i (1-e^{-\lambda}\lambda)^{t-i}\end{aligned} \tag{1.15}$$

Finally, the inverse transform is computed as:

$$P(a|n,t) = \frac{n!}{t^n}[\lambda^n]\{e^{\lambda t}\boldsymbol{P}(\lambda,t)\} \tag{1.16}$$

with $[\lambda^n]\{h(\lambda)\}$ denoting the coefficient of λ^n in the power series $\{h(\lambda)\}$. So, we have to rewrite Eq. (1.15) as a power series in λ and extract the appropriate coefficient. We use first the binomial expansion $(a+b)^c=\sum_{k=0}^{h}\binom{h}{k}a^k b^{h-k}$

$$P(a|n,t) = \frac{n!}{t^n}[\lambda^n]\{\binom{t}{i}\sum_{c=0}^{t-i}\binom{t-i}{c}(-1)^c e^{\lambda(t-i-c)}\lambda^{c+i}\} \tag{1.17}$$

and using the expansion of the exponential function as a power series, the sum in Eq. (1.17) can be rewritten as:

$$\begin{aligned}&\sum_{c=0}^{t-i}\binom{t-i}{c}(-1)^c\sum_{j=0}^{\infty}\frac{(t-i-c)^j}{j!}\lambda^j\lambda^{c+i}=\\&=\sum_{j=0}^{\infty}\lambda^{j+i}\left(\sum_{c=0}^{j}(-1)^c\binom{t-i}{c}\frac{(t-i-c)^{j-c}}{(j-c)!}\right)\end{aligned}\tag{1.18}$$

and extracting the coefficient of λ^n for the appropriate n value, $n = j + i$, we obtain the result in Eq. (1.8).

Chapter 2

OSL: An Optimization-based Scheduler for RFID dense reader environments

– Who are you?
– I am Death.
– Have you come for me?
– I have long walked by your side.
– So I have noticed.
– Are you ready?
– My body is ready, but I am not.

2.1. Introduction

In the previous Chapter, the scheduling problem for generic single-channel environments (f=1) is studied. A mathematical model to optimize the number of identifications was developed, however the contraints imposed were too strict and led to suboptimal resource space reutilization.

In the current Chapter we extend these results to the most general case (arbitrary m, f, t parameters and arbitrary topology). In this case, competing readers can be multiplexed either in time or frequency.

The problem is how to distribute the reading resources available among the readers to perform optimally. Our goal is to select for each reader a working frequency and a set of slots where they can operate without interferences, such that *the expected number of identified tags in the whole network is maximal*. The key aspect of our model is incorporating available knowledge about the load (number of competing tags) of each reader to decide the schedule. Previous optimization models (see Section 2.2) have largely ignored the availability of this information, but it can be effectively used in an scheduler.

Henceforth, we have developed a new mathematical model for the system denominated OSL: An Optimization-based Scheduler for RFID dense-reader environments, which leads to an integer non-linear non-convex optimization problem (NP-hard). We have tackled it using a stochastic solver based on parallel Simulated Annealing [Kirkpatrick et al., 1983]. Although this solver is not able to guarantee finding a global optima (neither any other existing solver), the solutions obtained (see Section 2.6) outperform current state-of-art RFID schedulers. Moreover, this new approach allows the maximal resource space reutilization overcoming the problem of previous model in the previous Chapter.

In Section 2.2 the most relevant research proposals are shown. Section 2.3 describes the FSA anti-collision procedure. Section 2.4 describes the mathematical base of OSL, and Section 2.5 its implementation. Section 2.6 studies its performance and, finally, Section 2.7 concludes and describes future works.

2.2. Related work

Many new proposals for coordinating dense reader environments have been proposed recently such as [Ferrero et al., 2012], [Eom et al., 2009], [Gandino et al., 2013], or [Li and He, 2013]. These heuristics are usually aimed at improving "throughput", defined as the average percentage of utilization time of the channel without reader-to-reader collisions. State-of-the-art methods like DCNS [Gandino et al., 2013] or NFRA [Eom et al., 2009] operate selecting a "color" for each reader. Each scheduling period is divided into time slot blocks, each one associated to a particular color. Readers with a given color can transmit using only the corresponding blocks. The final performance depends on the number of colors, which algorithms adjust dynamically using different criteria, and on the way algorithms "solve" collisions, *i.e.* how colors are reassigned after a collision event. Besides, another cause of inefficiency of these heuristics is that some "colors" (*i.e.* time blocks) may remain unused if none reader selects them. In contrast, based on the load information, OSL aims at maximizing the number of identifications instead of the utilization ratio, which is maximal in OSL since neither collisions nor empty slots can occur. Besides, belonging to this family of heuristics [Ferrero et al., 2012] also consider the number of tag readings, assuming a constant figure of 100 tags read in 0.46 s. However, this information is not incorporated in the model to decide the length of the scheduling period assigned to each reader, which remains constant, contrary to OSL.

Besides, some works use physical-layer models and aim at reducing the interference level, *e.g.* [Gong et al., 2012], [Choi and Lee, 2011], [Tanaka and Sasase, 2007] or [Kim et al., 2007]. Unlike OSL, none of these works consider the impact of the reading protocol or the load information.

On the other hand, some other proposals consider information directly related to the identification procedure. [Deolalikar et al., 2005] derive in optimal scheduling schemes for RTI single-channel RFID networks. As in OSL, authors aim at maximizing the number of identifications within the scheduling period, but they consider an oversimplified tag-reading model. On the contrary, OSL uses an exact computation for the tag-reading process (Section 2.3). [Seo and Lee, 2010] propose a scheduler based on a RRI model, which seeks to maximize a utility function depending on the assigned time slots, but without load information or suitable anti-collision model. Besides, [Mohsenian-Rad et al., 2010] design two optimization-based algorithms: FDFA (which is fully distributed and achieves a local optimum) and SDFA (semi-distributed and reach to the global optimum). Similarly to OSL, they consider a FDMA/TDMA scheduler, but with the goal of achieving max-min fairness in the network, distributing resources evenly among all readers. However, neither of these proposals consider load information.

Finally, other works such as [Alcaraz et al., 2013] develops schedulers when mobility of the tags is considered, but in this case the goal is limiting *Tag Loss Ratio* (the ratio of tags that leave the system unidentified) in the network, rather than maximizing the number of identifications, as OSL and many the previous schedulers intend. Let us remark that readers can easily control tag mobility by stopping movement (*e.g.* stopping a conveyor belt) while tag-to-tag collisions are detected, *i.e.*, while there are tags still unidentified.

Summarizing, previous optimization models ignore the operation of the underlying anti-collision protocol and the availability of load information about the number of tags within range of each reader. However, this information can be very effectively exploited by a scheduler, as we show in Section 2.6.

2.3. FSA tag identification procedure

Let us provide a summary of FSA identification procedure (as explained in 1.2.1). RFID readers energize and start tag interrogation by issuing *Query* packets. To reduce tag-to-tag collisions, tags implement some anti-collision protocol such as FSA, which is the most prevalent in the UHF band (standardized in EPCGlobal Class-1 Gen-2 [EPCglobal, 2013]). In FSA, time is divided into time units called *slots*, which are grouped in *frames*. The number of slots composing a frame (k) is indicated in the *Query* packet. In FSA each tag randomly selects a slot within the frame to transmit, resulting in an identification (if only one tag answers in that slot), a collision (if multiple tags answer) or an empty slot. Correctly identified tags withdraw from the identification process.

There are two main variations of FSA, Static-FSA (with frames of constant length), and Dynamic-FSA which enhances performance by balancing the number of tags and slots in the frame (avoiding excessive collisions or empty slots). DFSA

operation is optimal in terms of reading throughput (rate of identified tags per slot) when the frame length equals the number of contenders [Vales-Alonso et al., 2011]. Therefore, to maximize the throughput the reader should know the actual number of competing tags. This number can be estimated based on observed statistics like the number of empty slots or collisions by using different methods, *e.g.* maximum-likelihood estimation [Vales-Alonso et al., 2011]. Some estimators have been also proposed in the context of dense-reader environments [Shah-Mansouri and Wong, 2011], [Shahzad and Liu, 2014] (see also Section 2.5.3).

A scheduler will assign a set of (possible non-consecutive) slots to each reader (see Figure 2.1). Let us note that in FSA, even if the slots assigned to a reader are non-consecutive, they can be considered as a compact block, since a tag does not proceed with the next slot of the identification frame until a *QueryRep* packet arrives from the reader [EPCglobal, 2013]. Therefore, during non-assigned slots tags are just waiting next command to continue the identification process[1]. Henceforth, let us denote as t_j the number of slots allocated by the scheduler to reader j. During these t_j slots a number of reading frames can be allocated (see Figure 2.1).

As was shown in 1.3.1 and 1.3.2, given the number of contending tags (n_j) for a reader j it is possible to compute the expected number of tag identifications, $\varphi(n_j, t_j)$, under SFSA and DFSA. Its computation is summarized next:

2.3.1. $\varphi(n_j, t_j)$ computation for SFSA

In the SFSA case, the reading process for each reader j consists of several consecutive reading frames of length k, until all the t_j reading slots are eventually exhausted. It is assumed that t_j is a multiple of k. Since expectation is a linear operator, $\varphi(n_j, t_j)$ can be computed as the sum of the average number of tags identified in the first frame ($\varphi(n_j, k)$) plus those identified in the remainder process ($\varphi(n_j - \Lambda, t_j - k)$), where Λ denotes the random number of tags identified in the first frame. In 1.3.1 is shown that this leads to the recursive formula (2.1), where $P(\lambda|n_j, k)$ is the probability of identifying λ tags if n_j tags contend in a single frame of k slots. This probability is computed in Appendix A in Chapter 1.

$$\varphi(n_j, t_j) = \begin{cases} \sum_{\lambda=0}^{n_j} (\lambda + \varphi(n_j - \lambda, t_j - k)) P(\lambda|n_j, k), \text{ if } t_j \geq k, \\ 0, \text{ otherwise} \end{cases} \tag{2.1}$$

2.3.2. $\varphi(n_j, t_j)$ computation for DFSA

In the DFSA case, the reading process for each reader j also consists of several reading frames but of variable length $k_{j,1}, k_{j,2}, \ldots$, until all the t_j reading slots are

[1] Readers must continue energizing tags to avoid they leave identification process, *i.e.* transmitting the Continuous Wave (CW) signal, which does not produce RRI since it is out-of-band with tags' backscattered signals [EPCglobal, 2013].

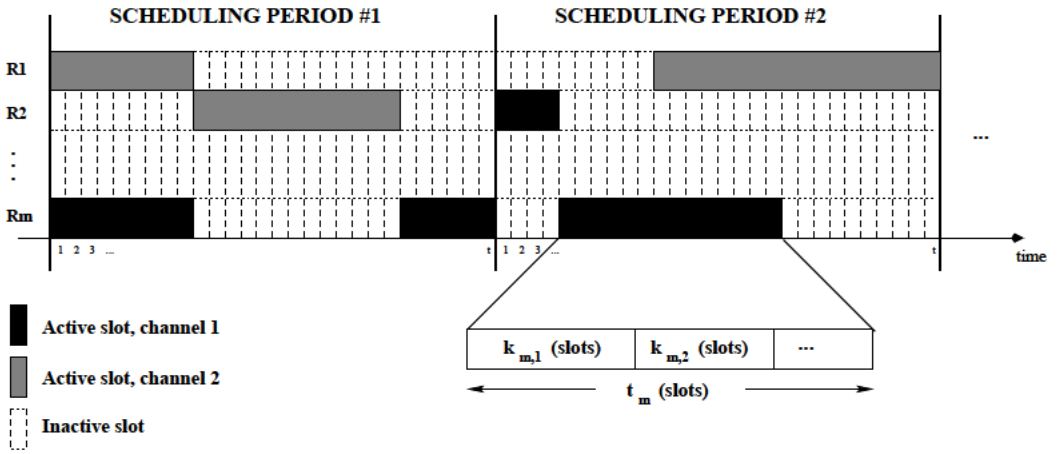


Figure 2.1: Example of scheduling in a RFID dense reader environment with f=2. At the beginning of each scheduling period the operation channel and the slots assigned are decided, changing from period to period. Note that the transmission period allocated to each reader is further subdivided in reading frames where the corresponding tag-to-tag anti-collision protocol is run into

used. Besides, the number of contenders in each frame is $\nu_{j,1} = n_j, \nu_{j,2}, \ldots$. We assume that DFSA tries to maximize reading throughput by allocating as many slots as the number of contending tags ($k_{j,i}$=$\nu_{j,i}$). This is possible while the remainder number of slots is greater than the number of contenders. Otherwise we simply assume that a last frame i^* is allocated with all the remaining slots ($k_{j,i^*} = t_j - \sum_{c=1}^{i^*-1} k_{j,c}$). Similarly to the previous case, $\varphi(n_j, t_j)$ can be described recursively by the formula (2.2) (see 1.3.2).

$$\varphi(n_j, t_j) = \begin{cases} \sum_{\lambda=0}^{n_j} (\lambda + \varphi(n_j - \lambda, t_j - n_j)) P(\lambda | n_j, n_j) \text{ if } n_j < t_j \\ \sum_{\lambda=0}^{n_j} \lambda P(\lambda | n_j, t_j) \text{ if } n_j \geq t_j \end{cases} \tag{2.2}$$

2.4. OSL - Optimization-based Scheduler with Load information

In this section we describe the mathematical basis of OSL in depth. In Our model (see Figure 2.1) we assume that time is divided into slots which are grouped into *scheduling periods* of length t slots. At each one of these periods, the goal is to identify the maximum number of tags distributing the available resources (f channels and t slots per channel) among the m readers. For each reader j=$1, \ldots, m$, we assume that the number of tags competing in its reading area at the beginning of the scheduling period, n_j, is known (see Section 2.5.3).

Besides, the existence of RRI among each pair of readers is represented by an $m \times m$ adjacency matrix A=$(a_{jj'})$ whose elements are 1 if reader j and j' can

not operate at the same time (*i.e.* readers j and j' interfere each other), and 0 otherwise (by definition a_{jj}=0 for all j=1, ..., m). See Section 2.5.3 about topology computation.

The scheduling must assign a set of (maybe non-consecutive) slots on a channel for each reader j. We consider the operational constraint that a reader remains in the same channel during the whole scheduling period [2]. Then, a *scheduling* or *solution* is an $m \times t$ matrix S such that:

1. Element s_{jl} indicates the channel, $f_j \in \{1 \dots f\}$, assigned to reader j on slot l, when the reader is active. If the reader must remain inactive, then s_{jl}=0. Note that f_j=0 if all slots in the scheduling period are inactive.

2. Neighbor readers do not operate simultaneously in the same channel, *i.e.* $s_{jl} \neq s_{j'l}$ for all $l \in \{1, \dots, t\}$ and for all jj' such that $0 \neq f_j = f_{j'} \neq 0$.

2.4.1. Normal Form

The aim is at finding solutions maximizing the overall throughput. Let us recall that $\varphi(n_j, t_j)$ denote the expected number of identified tags when n_j tags contend in t_j slots, and let us denote as Φ the expected number of identified tags in the whole network during a scheduling period, that is,

$$\Phi = \sum_{j=1}^{m} \varphi(n_j, t_j) \tag{2.3}$$

The number of slots allocated to a reader j is $t_j = \sum_{l=1}^{t} 1_{s_{jl}>0}$, where $1_{s_{jl}>0}$ is an indicator function, having the value 1 for all elements $s_{jl} > 0$ and the value 0 for all elements $s_{jl} = 0$. It leads to:

$$\Phi = \sum_{j=1}^{m} \varphi(n_j, \sum_{l=1}^{t} 1_{s_{jl}>0}) \tag{2.4}$$

Then, the optimization problem can be stated as: Given parameters f, t, m, an adjacency matrix A, and the number of contenders per reader $\{n_j\}_{j=1,\dots,m}$, solve:

$$\max_{\substack{(\{s_{jl}\}) \\ j=1,\dots,m \\ l=1,\dots,t}} \Phi \tag{2.5}$$

[2] The frequency switching delay in readers is noticeable, e.g., in the order of several hundred milliseconds [Mohsenian-Rad et al., 2010], therefore channel switching is impractical.

Subject to:

1. $s_{jl} \in \{0 \dots f\}$
2. $s_{jl} = s_{jl'}$ if $s_{jl} > 0$ for $l' = 1, \dots, t$
3. $s_{jl} \neq s_{j'l}$ if $a_{jj'} = 1$ and $s_{jl} > 0$ for $j' = 1, \dots, m$

for all $j = 1, \dots, m$, and for all $l = 1, \dots, t$.

Constraints 1) to 3) are direct translations of conditions previously imposed on the definition of solutions. This problem is a Non-Linear Non-Convex Program with integer variables subject to non-linear constraints, and therefore is NP-hard. Known methods able to find global optima for integer convex non-linear programs exist, such as Branch and Bound [Land and Doig, 2010] or Extended Cutting Plane (ECP) [Westerlund, 1992], but in our problem convexity conditions are not met. Thus, these methods will behave as heuristics, *i.e.* without any guarantee of global optimality. Indeed, due to the large amount of variables (mt) and constraints involved ($mt[1 + t + m]$), the use of the aforementioned methods to address our optimizacion problem is computationally infeasible using state-of-the-art solvers like Alpha-ECP (based in ECP) or BARON (based in Branch and Bound) [Brooke et al., 1998].

An alternative and effective solution has been found using a stochastic method based on parallel simulated annealing and developed specifically to target this problem. Its operation and performance are discussed in Section 2.5.1.

2.4.2. Compact Form

For SFSA protocol, a simple way to reduce the complexity of our optimization program is to group slots into elemental assignment units, with the size of a frame (k). That is, instead of deciding the assignment slot-by-slot, it is decided frame-by-frame. Formally, the aim is at finding a *compact* form S' of S such that element $s_{jr} = s'_{jr'}$ for $r = r'l, r'l + 1, \dots, (r' + 1)l - 1$, and where the constraints for S' are identical to the constraints considered for S. Note that in this case *element* refers to a frame rather than to a slot, as in the previous section.

The optimization problem is expressed as: Given parameters f, t, m, k, an adjacency matrix A, and the number of contenders per reader $\{n_j\}_{j=1,\dots,m}$, solve:

$$\max_{\substack{(\{s'_{jr}\}) \\ j=1,\dots,m \\ r=1,\dots,t/k}} \sum_{j=1}^{m} \varphi(n_j, k \sum_{r=1}^{t/k} 1_{s'_{jr}>0}) \tag{2.6}$$

Subject to:

1. $s_{jl} \in \{0 \dots f\}$
2. $s'_{jr'} = s_{jr''}$ if $s_{jr'} > 0$ for $r'' = 1, \dots, t/k$
3. $s'_{jr'} \neq s_{j'r'}$ if $a_{jj'} = 1$ and $s_{jr'} > 0$ for $j' = 1, \dots, m$

for all $j = 1, \dots, m$, and for all $r' = 1, \dots, t/k$.

With this compact formulation both the variables and the constraints are reduced by a factor k, leading to simpler problems. Let us remark that this alternative reduces the exploration space, discarding solutions where slots are not grouped into sets of length multiple of k. Clearly, these discarded solutions would be suboptimal, for example, if t_j=20 and k=16, then only one reading frame of 16 slots can be allocated and 4 slots are lost.

For the DFSA case, a similar strategy can be considered, but in this case it is not true that all the solutions discarded would be suboptimal, and therefore configurations based on DFSA must be solved using the normal form.

2.5. OSL implementation

2.5.1. Simulated annealing solver

Our solver is based on the simulated annealing meta-heuristic, which is successfully used in many practical optimization problems in networking [Cao et al., 2013]. Our implementation is multi-threaded, where each thread continuously seeks for new solutions. As conventional annealing, solutions are accepted either if they improve the global best known solution, or (if solution does not improve) based on a probabilistic event that is more likely for higher algorithm "temperatures". Precise operation details and parametrization of the annealing are given in [Vales-Alonso et al., 2003].

The key aspect of our implementation is the method to generate new solutions. It guarantees that all the solution space can be explored, and that solutions generated are always feasible, *i.e.* that all constraints of the problem (2.5) or (2.6) are fulfilled. Algorithm 1 overviews how this method works. Basically, given a solution S, an element[3] s_{jl} is selected randomly, and a new frequency channel is selected (which must be the same frequency used by the rest of active elements in the row, if any, or otherwise one picked at random). If the element s_{jl} is already active, then another element is selected until eventually an inactive one is found (if all elements are active the solution can not be improved and the algorithm ends).

[3] As stated in Section 2.4 an element represents a single slot in the normal form, or a group of k slots in the compact form.

Algorithm 1 Solution generation

```
#   Parameters:
#   S (matrix m × t), previous solution
#   A (matrix m × m), network topology
#   Main internal variables:
#   channel (vector size m), frequency channel assigned to reader j
#   active (vector size m), indicates whether reader j is active in the new solution
#   order (vector size m), indicates the order in which readers are traversed

for j=1:m do
    channel[j] ← max_{l∈{1,...,t}} S[j][l]
end for

# Can solution be improved?
if Σ_{j=1}^{m} Σ_{l=1}^{t} 1_{S[j][l]} == mt then return S
end if

# Search for some element that can be activated
repeat
    row ← U[1, m]; column ← U[1, t] # Random position to change in the solution matrix. U[a, b] returns an uniformly random integer value between a and b
until S[row][column] ≠ channel[row] OR channel[row]==0

# Select operation frequency for the element in the freq variable
freq ← channel[row]
if channel[row]==0 then
    freq ← U[1, f] # Returns an uniformly random integer value between 1 and f
end if

# Search and apply interdependences to readers in the same channel or inactive
for j=1:m do
    active[j] ← true
end for

order ← [row, PERM(1,...,row-1,row+1,m)] # Random permutation of the readers order with the reader 'row' always in first place
for aux=1:m do
    j ← order[aux]
    if active[j] AND (channel[j]==freq OR channel[j]==0) then
        S[j][column] ← freq # Element gets activated
        for aux'=aux+1:m do
            j' ← order[aux']
            if A[j][j']==1 AND (channel[j']==freq OR channel[j']==0) then
                active[j'] ← false
                S[j'][column] ← 0 # Element gets deactivated
            end if
        end for
    end if
end forreturn S
```

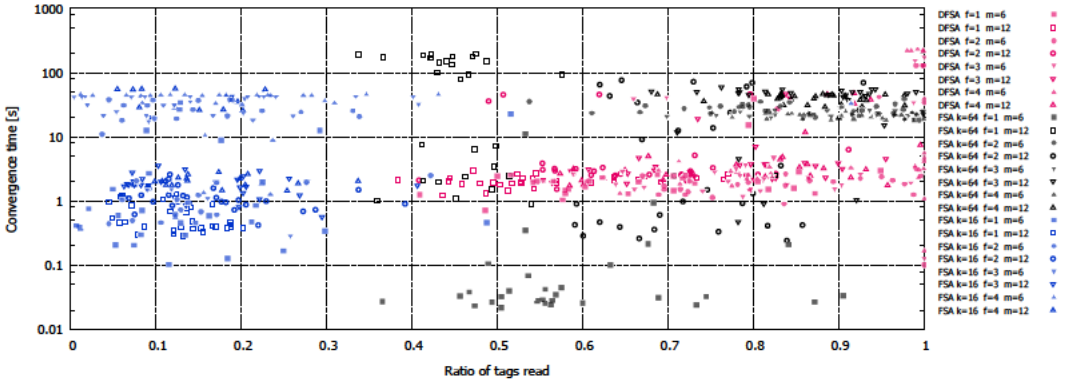


Figure 2.2: Annealing algorithm convergence speed. Experiments have been run on an AMD Opteron@2600 MHz 48 multi-core system

The activation of the selected element may affect other elements on the column. Some must be deactivated to fulfill constraints, and others (due to possible deactivations) could become also activated to improve the solution. The particular sets of elements to activate and deactivate depend on the order considered for the rows, thus it is selected at random to guarantee that the algorithm can explore all possible combinations. In addition to this basic solution generation algorithm, the annealing also creates new solutions deactivating all elements on a row or on a column. Although this is not strictly necessary, it speeds up algorithm convergence. Finally, the algorithm can be initiated with any valid solution (*e.g.* $S \leftarrow 0$, in our tests).

2.5.2. Convergence analysis

Different tests have been performed to measure the quality and the speed of our algorithm. All the scenarios described in [Vales-Alonso et al., 2013c], which are limited to a single channel (f=1), have been also analyzed with the optimization model developed in this work and solved by means of the annealing algorithm. Results are better than those reported in [Vales-Alonso et al., 2013c] if an improved space reutilization is possible (*e.g.* Star scenario in next section), or coincident otherwise.

As stated previously there is no guarantee that the simulated annealing metaheuristic obtains a *global* optimum. This kind of combinatorial problems have multiple local optima. The annealing performs well in many cases since it tries to "escape" from them to find the global one. For simple cases, where the cardinality of the search space is low, *e.g.* Star scenario in Section 2.6, it can be exhaustively checked to find the global optimum. In all the test performed for these simple cases the annealing found the global optimum.

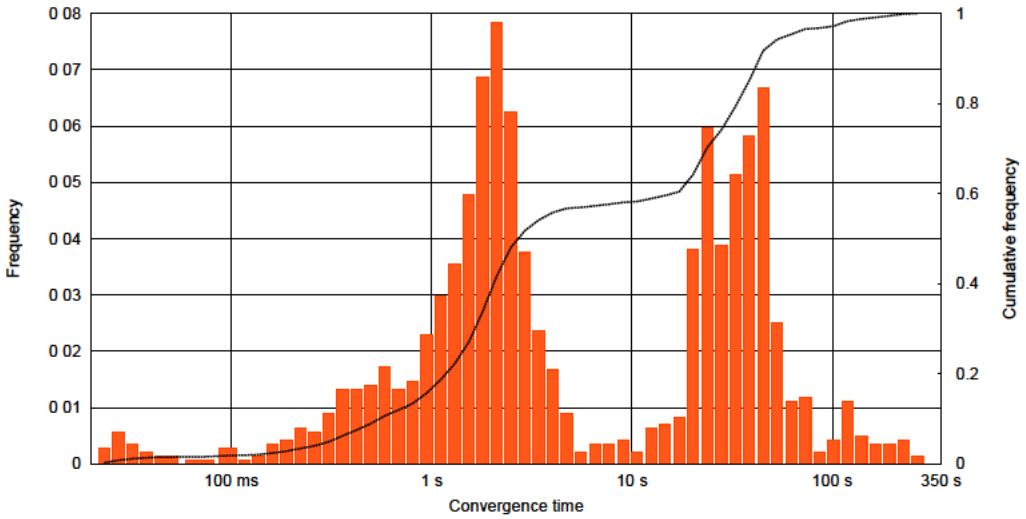


Figure 2.3: Annealing algorithm speed performance. Experiments have been run on an AMD Opteron@2600 MHz 48 multi-core system

Regarding the convergence speed, Figure 2.2 shows the result of a basic experiment: 30 topologies of $m = 6$ and $m = 12$ readers have been generated randomly, and for each one the converge time has been obtained and plotted versus the ratio of tags read, that is, versus $\gamma = \sum_{j=1}^{m} n_j/\Phi$. In the experiments n_j are selected at random uniformly in the interval $\{1, \ldots, 200\}$. Intuitively, the ratio γ indicates the global "suitability" of the problem. If it is close to 0, a minority of the tags can be read since there are not enough resources to read all tags and then the scheduling problem may become too complex. On the contrary, ratios close to 1 reveals that enough resources exist for the scheduler to produce good solutions. In the figure, it can be clearly seen that the tag-to-tag anti-collision protocol selected determines the ratio γ of the problem, separating the points in three distinctive "clouds" ('blue', 'red' and 'black' corresponding to each of the tag-to-tag anti-collision protocols). Besides, the convergence time seems to be mainly determined by the size -number of readers- of the problem (m). Another important result shown is that for Static-FSA k=16 configurations the performance achieved is always very low compared to Static-FSA k=64 or Dynamic-FSA, as discussed previously.

Figure 2.3 shows an histogram for the result of next experiment: 30 topologies of m=6, and 30 of m=12 readers have been generated randomly, measuring the convergence time. More than 90% of the tests required less than 100 s, and all of them converged in less than 350 s. We should also remark that about 50% of the cases required less than 10 s. The two peaks in the convergence speed in Figure 2.3 seems motivated by the underlying protocol, as well as by the number of channels available. The peak at the left (less convergence time) corresponds mainly to DFSA or FSA combinations with a single channel, while the other comprises most of the remaining combinations.

2.5.3. Practical use of OSL

To apply OSL, we consider two phases:

1. Preprocessing, consisting of (i) Topology computation, and (ii) Scheduling computation. This phase is required since they can not be performed in real-time.

2. Operation, consisting for each scheduling period in (i) Load determination, and (ii) Application of the scheduling.

During preprocessing, the topology of the particular dense reader environment must be obtained first. We assume that readers are installed in fixed positions and thus topology does not vary (see Section 2.5.5 about operation in dense reader environments with mobile readers). In this case, topology can be determined either *off-line* by measuring the RRI levels, or *on-line*, *e.g.* authors of [Mohsenian-Rad et al., 2010] propose the periodical exchange of HELLO messages over control channels at pre-determined transmission powers to construct the topology. Note that readers with several transmission antennas are considered as a single element in the topology, since they operate simultaneously. Besides, RRI must be computed in the reception antennas, which can differ from the transmission ones in case of bistatic readers.

When the topology is known, the optimal scheduling must be computed. However, this cannot be performed in real-time as indicated in the results of the previous section (typical solution computation requires a period of seconds which is in the order of a practical scheduling period length). Besides, solutions must be obtained for each possible load configuration of the system. Although this process is costly, it has the advantage that solutions do not change as long as there are no changes in the network.

After preprocessing phase, the system executes the operation phase. At the beginning of each SP, the load information is obtained by each reader using some tag-count estimators like e.g. [Shahzad and Liu, 2014] which has been developed to detect rapid changes in the tag count and work in dense reader environments. The load information is transferred to a central coordinator or directly distributed to all other readers. Then, scheduling decisions are made by simply searching into the appropriate precomputed lookup table, which is extremely fast and do not add computational load to the process. This table contains, for each reader, the operation channel and the precise time slots where OSL allows transmission (*i.e.* the information shown in Figure 2.1 or in the examples provided in Section 2.6). Note that communications occur only at the beginning of each SP, and require a minimal amount of data transmission: the load information from each reader. Therefore, communications do not hinder the operation of the RFID network.

2.5.4. Synchronization issues

Since OSL requires that all readers begin slots at the same moment, a tight synchronization is necessary. Typically RFID slots last in the order of milliseconds [Floerkemeier and Sarma, 2009] (although this may depend on the exact modulation or packets format), but they do not start until a *QueryRep* is issued by readers. To implement this scheme, a synchronization protocol must be used. Typical wired network synchronization protocols achieves an accuracy in the order of 1 μs [Leong et al., 2006], while average errors of 4.8 μs have been reported in wireless systems [Suyoung and Sichitiu, 2005]. A safe solution to the slot synchronization issue is to have a small guard period before starting each slot. However, the impact in the performance is minimal ($< 1\%$ according to the data given above).

2.5.5. Dense reader environments with mobile readers

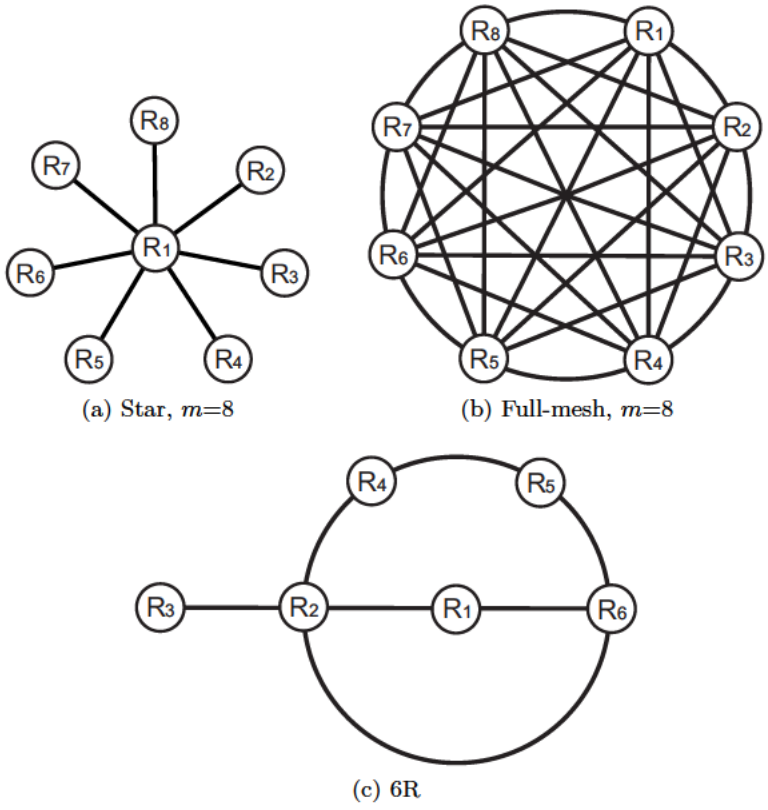
When mobile readers (MRs) are present, OSL is also able to operate assuming the worst-case scenario: a MR always causes RRI to any other reader, and *vice versa*. Scheduler is computed under this assumption ($a_{jj'} = 1$) in the adjacency matrix A either if reader j or j' is a MR. This assumption is motivated because a RFID MR may cause temporal and fast-changing RRI due to its unpredictable movement, and it would be impractical to determine RRI topology in real-time. Note that with this solution is not required to know MRs position, but just knowing which readers are mobile. In the next section we show that depending on the ratio of MRs and the probability of collisions, OSL is still able to perform better than state-of-art heuristics in a broad set of cases, even with this restrictive topology setup.

2.6. Results

Three representative scenarios have been selected for our tests and are represented in Figure 2.4. The first one, Star topology, represents a practical case where RRI is restricted to pairs (*e.g.* by using a-priori scenario arrangement) between the central reader R_1 and the other readers. Star topology allows frequency multiplexing between the central element and the other readers if $f > 1$. The second scenario, Full-mesh topology, represents a case where all pairs exhibit RRI, which may occur in many RFID installations where no counter-measures have been taken to reduce interferences. Finally, the 6R topology, is a six reader scenario with jumble links, where *a priori* it is difficult to select feasible solutions.

The following configurations have been considered:

- $f = 1$ to 4 channels,
- $t = 512$ slots per channel,



(a) Star, m=8 (b) Full-mesh, m=8

(c) 6R

Figure 2.4: Example scenarios

- n tags to be identified at each reader (*i.e.* there is a homogeneous load in the network), from 5 to 200 tags with a step of 5 tags,
- $m = 6$ for the 6R scenario,
- $m = 6$, 8, 10 and 12, for the Star scenario,
- $m = 6$ and 12, for the Full-mesh scenario,
- and for SFSA $k = 16$ and 64.

2.6.1. Comparison of OSL *versus* heuristics

Firstly, the performance of OSL has been compared against other schedulers. The selected for comparison was the DCNS protocol [Gandino et al., 2013]. DCNS is a multi-channel (as our proposal) distributed state-of-the-art heuristic and authors report a better performance than other existing approaches based on distributed coloring algorithms, such as NFRA [Eom et al., 2009], DCS, or ColorWave (see [Gandino et al., 2013] for a reference about these protocols). As stated in Section 2.2, these heuristics are aimed at improving "throughput", *i.e.* the ratio of time with

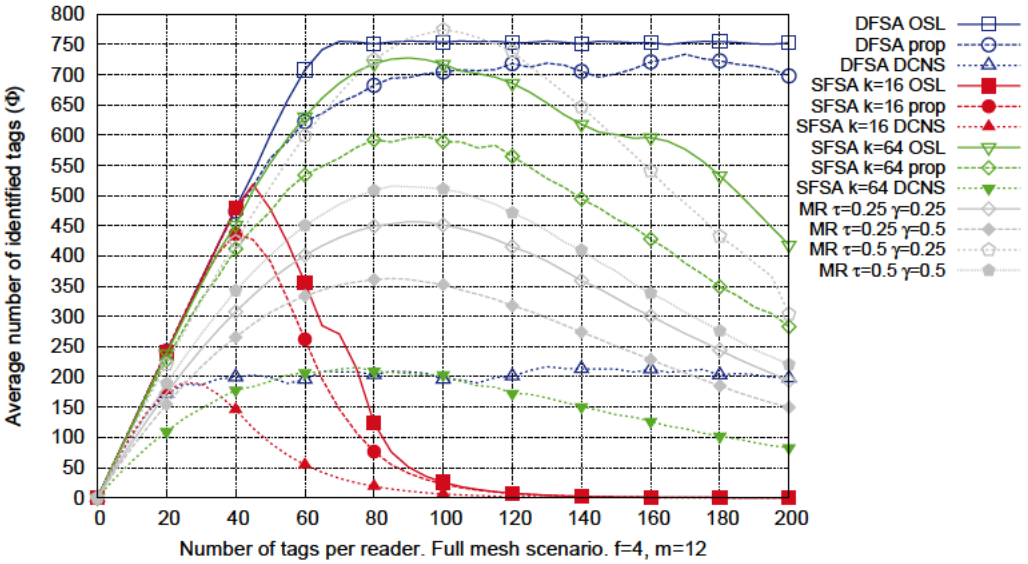
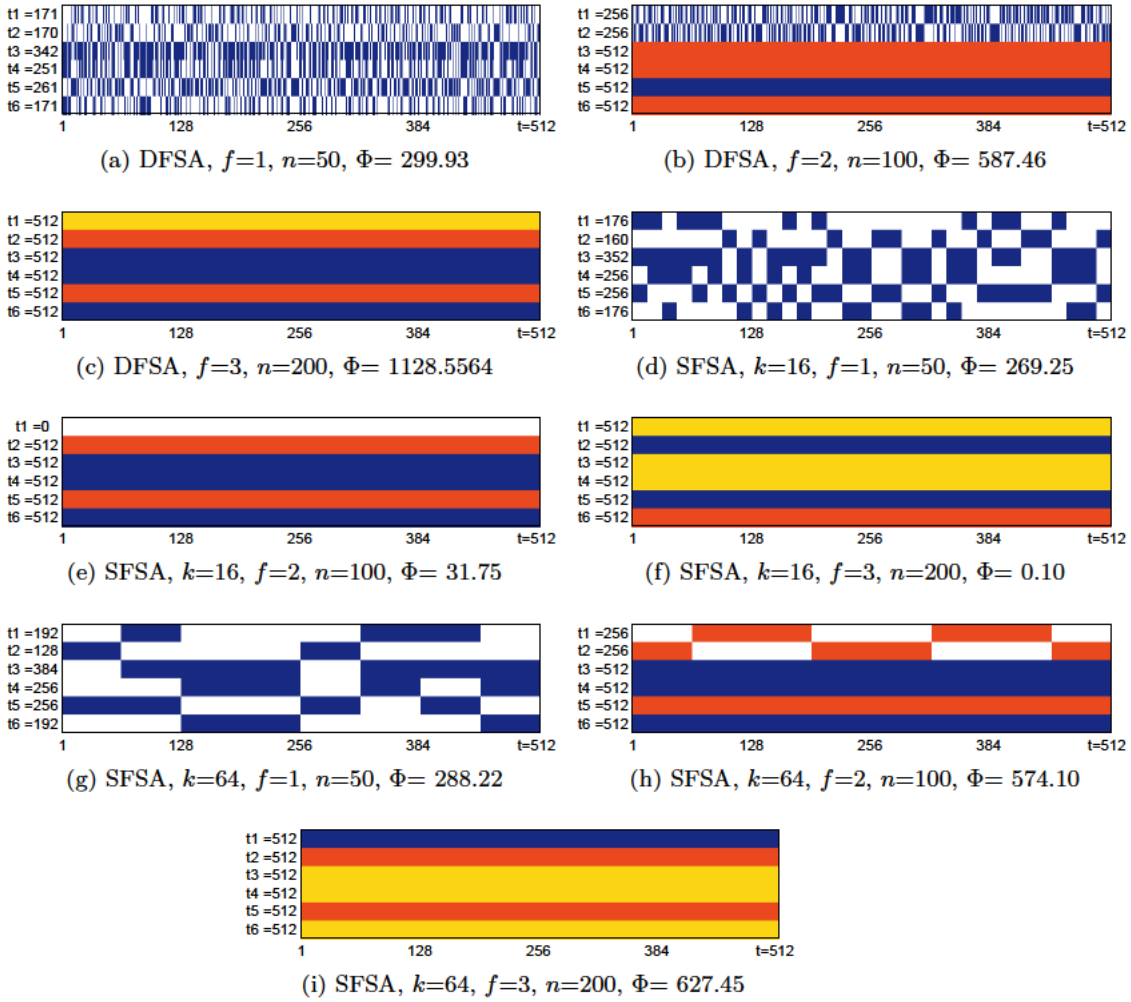


Figure 2.5: Optimal *vs.* proportional allocation, Full-mesh scenario, m=12, f=4

effective transmissions. Thus, utilization is reduced by collisions among readers, and by unassigned slots as well. In contrast, the throughput is maximal in OSL (100 %) since neither collisions nor unassigned slots can occur. Therefore, using throughput as a comparison measurement is unfair for other schedulers, and instead we are comparing the identifications per scheduling period in the network (*i.e.* Φ). This metric is also used in [Ferrero et al., 2012] to study the performance.

Figure 2.5 shows performance of DCNS *versus* OSL for the Full-mesh scenario. In DCNS the block sizes (see Section 2.2) are assumed to be the frame length for SFSA, while the result for DFSA is the best obtained using blocks of sizes power of 2 from 16 to 512. The DCNS parameters were set to $Upsafe = 0.6$, $Downsafe = 0.4$, and the *minimal iterations in a color* parameter was set to 10. Result is obtained averaging 10000 Monte Carlo iterations (discarding 1000 transient iterations to allow convergence to the best color allocation).

Results indicate a major improvement using OSL, mainly due to the avoidance of unassigned periods and collisions since OSL relies on a coordinated approach. Indeed, tests with other heuristics have led to similar overwhelming improvements by OSL caused by the aforementioned reasons. Clearly, this comparison is unfair since a coordination scheme could be easily implemented for heuristics avoiding these problems. This leads to the question of which is the upper-bound performance for such a coordinated heuristic in a DRE and how it will compare against OSL. Moreover, the load information should also be considered available to the heuristics. Distributed heuristics of the kind of NFRA or DCNS are mainly based on a fair sharing of the available resources. A coordinated mechanism implementing a similar concept is a *proportional* sharing scheme. That is, the resources should be assigned



(a) DFSA, f=1, n=50, Φ= 299.93

(b) DFSA, f=2, n=100, Φ= 587.46

(c) DFSA, f=3, n=200, Φ= 1128.5564

(d) SFSA, k=16, f=1, n=50, Φ= 269.25

(e) SFSA, k=16, f=2, n=100, Φ= 31.75

(f) SFSA, k=16, f=3, n=200, Φ= 0.10

(g) SFSA, k=64, f=1, n=50, Φ= 288.22

(h) SFSA, k=64, f=2, n=100, Φ= 574.10

(i) SFSA, k=64, f=3, n=200, Φ= 627.45

Figure 2.6: Graphical representations for the optimal solutions in the 6R scenario. Each row represents the assignment of slots to a reader, and each channel is represented with a different color scale. White color represents inactive slots. At the left the number of active slots t_j is indicated for each reader

proportionally to the load of each reader, This mechanism achieves an upper-bound performance compared to fair distributed schemes. Formally, this upper-bound heuristic can operate as follows for DFSA[4]:

1. Select a random order $\eta_1, \eta_2, \ldots, \eta_m$ for the readers.

2. Following this order, channels are assigned. For reader η_j, frequency channels are considered in order from 1 to f. If, for a given channel c, the reader does not conflict with any of the previous readers (*i.e.* $a_{\eta_j \eta_{j'}}$=0, for all $j' < j$ such that c=$f_{\eta'_j}$), that channel is selected for reader j (*i.e.* f_{η_j}=c). If no channel c fulfills such condition, f_j is selected at random uniformly in $\{1, \ldots, f\}$.

[4]For SFSA the procedure is similar but assigning elements of k slots instead.

3. Slots are assigned. First[5], for each reader j, the number of neighbor readers sharing the same frequency channel, s_j, is computed: $s_j = \sum_{j'=1}^{m} a_{jj'} 1_{f_j=f_{j'}}$, where $1_{f_j=f_{j'}}$ is an indicator function having value 1 is $f_j{=}f_{j'}$, and 0 otherwise. Then, the number of time slots is assigned proportional to s_j and to n_j, as $t_j{=}\lfloor \frac{n_j}{\sum_{j'=1}^{m} n_j} \frac{t}{s_j} \rfloor$. This assignment guarantees that t_j slots can be allocated to reader j within the shared t slots.

4. Finally, distribute unassigned slots (due to the rounding operation) at random among neighbors.

Figure 2.5 shows performance of the upper-bound heuristic (labeled as "prop"), *vs.* OSL. The former achieves a reasonably good performance as can intuitively be expected, however the allocation is clearly suboptimal compared against OSL. This shows that OSL is performing better than *any protocol based on fair resource distribution*, whether it is distributed (e.g. ColorWave, NFRA) or coordinated (upper-bound heuristic) and whether it uses load information or not.

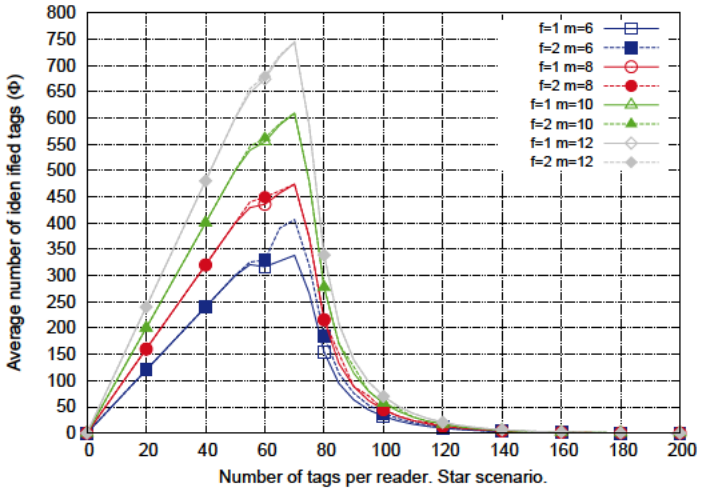
2.6.2. OSL performance analysis

Next, we discuss the performance of OSL for the scenarios described above. Figure 2.6 shows the optimal schedulers obtained for the 6R scenario for different example loads (n=50, 100, 200), following representation of Figure 2.1 where active slots are depicted in color while inactive slots are white. Clearly, the solutions are non-trivial in most cases (*i.e.* cannot be obtained through an educated guess) and depend on the scenario parameters. As expected, as f increases, the expected number of identifications Φ improves. Noteworthy, even with a similar allocation (e.g. f=3, n=200), the resulting Φ (shown in captions of Figure 2.6) vary largely among anti-collision protocols.

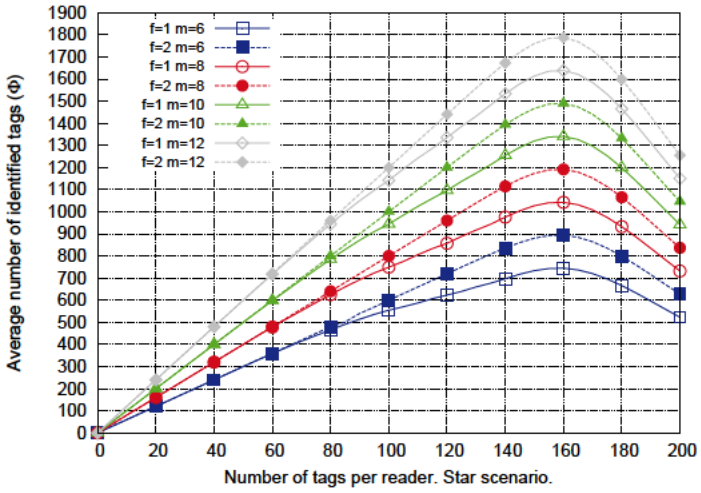
Figures 2.7 and 2.8 plot Φ *vs.* the load of the system for the Star and Full-mesh configurations. DFSA always performs better than SFSA assignments as can be expected, since DFSA achieves an optimal reading throughput frame-by-frame. Besides, DFSA performance stabilizes around an upper bound that depends on the available channels (this effect is clearly noticeable in Figure 2.8), whereas SFSA performance reaches a peak and then drops when n increases. Notably, it is even possible that the reader block as shown in Figures 2.7 and 2.8 for lower k values as 16.

Another important effect is related to the performance *vs.* the available channels. For example, in the Star scenario increasing the channels beyond $f = 2$ is useless. There is a partition of nodes in two subsets that achieves maximal performance assigning to each one a frequency channel. This same effect occurs for the 6R case for f=3, but it is not present in the Full-mesh case up to $f = 4$. If the

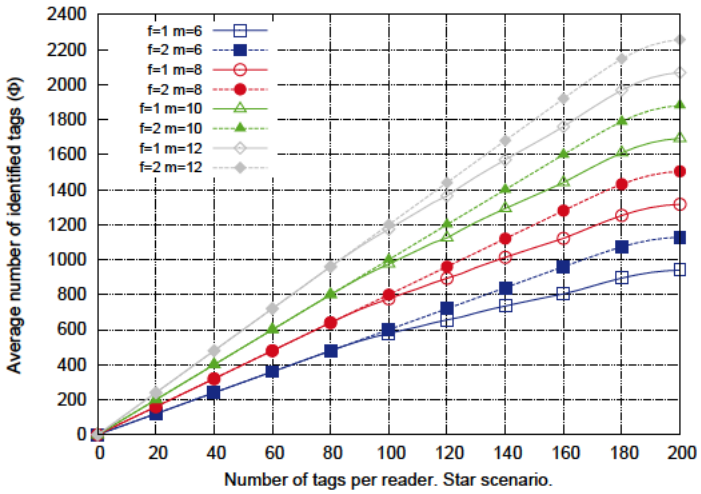
[5]In this step, readers can be traversed in any order.



(a) SFSA, k=16

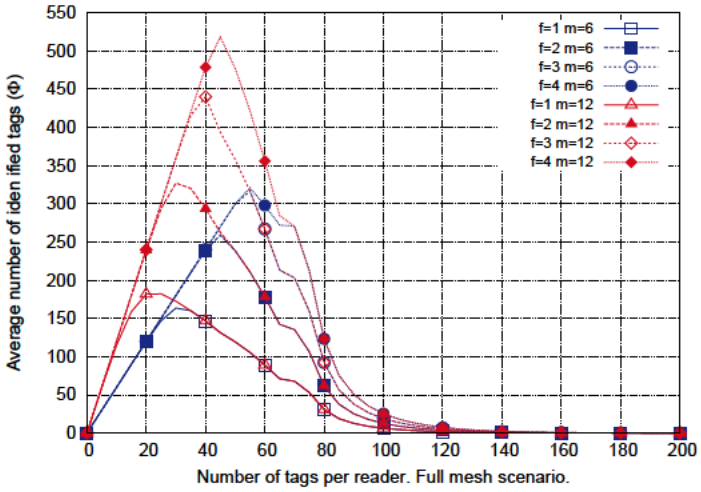


(b) SFSA, k=64

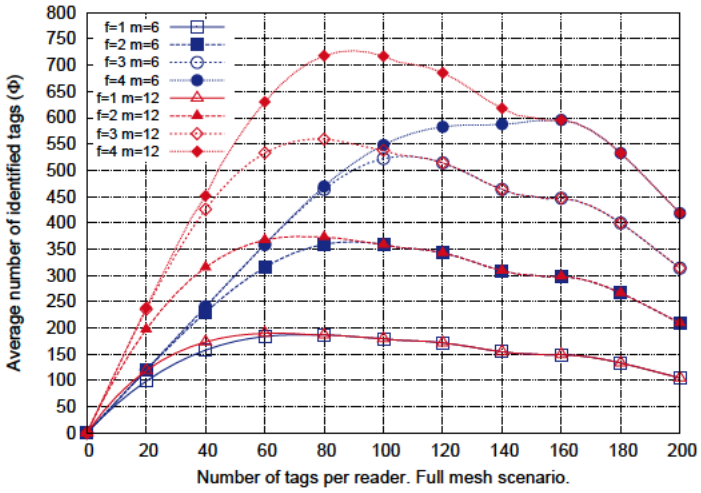


(c) DFSA

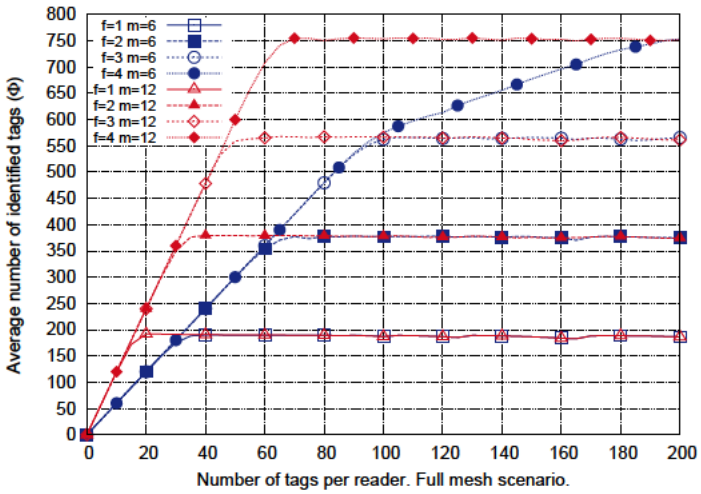
Figure 2.7: Expected number of identifications (Φ) vs. n. Star scenario, m=6, 8, 10, 12



(a) SFSA, k=16



(b) SFSA, k=64



(c) DFSA

Figure 2.8: Expected number of identifications (Φ) vs. n. Full-mesh scenario, m=6, 12

number of channels available is not enough, the scheduler divides time slots among interfering readers. In some cases, the scheduler may even decide to "deactivate" readers, *e.g.* in the Star scenario with f=1, there is a point ($n \geq 70$) where the best option is to disconnect the central reader and allocate 512 slots to each the remainder readers. This leads to maximal performance at the expense of *unfairness* on reader $R1$. Higher loads cause a higher unbalancing due to this prioritisation effect.

2.6.3. Performance in DREs with mobile readers

Finally, an experiment to analyze performance in dense reader environments with MRs has been conducted. We have considered a network with m=12 readers, a ratio of τ MRs and SFSA k=64 protocol, where the fixed readers have RRI among them. For this case, the performance of OSL corresponds to the Full-mesh scenario m=12 since, as discussed in 2.5.5, OSL assumes RRI for all the MR links. As comparison, the performance in DCNS is evaluated assuming that a probability γ models the existence of RRI in a MR link. Note that MRs are not screened and may easily affect other readers, so γ can be high. In our tests we have set τ=0.25, 0.5, and γ=0.25, 0.5. As shown in Figure 2.5, OSL (SFSA k=64 OSL) performs better than DCNS (plots labelled as MR) for most configurations. However, as can be expected, DCNS is better in cases where most readers are mobile (high τ) or they very seldom cause RRI (small γ).

2.7. Conclusions

This Chapter introduced OSL, a general multi-channel scheduler for dense reader environments. OSL is formulated to obtain maximal space resource reutilization and to avoid mutually interfering readers operating simultaneously on the same frequency channel. A practical solver to tackle the optimization problem is implemented by means of a parallel simulated annealing. The solutions obtained can be used to implement a system based on look-up tables computed *off-line* and, more important in our opinion, establishes the performance limits achievable. Therefore, results should be useful for evaluation of real-time heuristics. As a future work we aim at developing optimal schedulers with support of fairness criteria and priority among readers. Another critical topic is studying the dynamics of such systems.

Chapter 3

Analytical computation of the mean number of tag identifications during a time interval in FSA

– I'll see you in hell.
– Tell 'em Freddy sent ya.

3.1. Introduction

As seen in previous chapters, Framed Slotted Aloha (FSA) is the prevalent anti-collision protocol in the UHF band for Passive Radio Identification (RFID) systems, and has been standardized in EPCGlobal UHF Class 1 Generation 2 - ISO 18000-6C [EPCglobal, 2013]. In FSA, reading cycles (frames) begin with a *Query* packet. These frames consist of K time slots (or slots for short) delimited by *Query Rep* packets. This frame-length parameter k is indicated in the *Query* packets, and it is chosen by the reader depending on the protocol, the scenario constraints, and on variables of the identification process. In the EPC standard, K must be in the set $2^Q : Q \in \{0, 1, \ldots, 15\}$, leading to two basic alternatives: Static-FSA (SFSA) where k is fixed, and Dynamic-FSA (DFSA) where k is adjusted adaptively. Then, at the beginning of the frame each unidentified tag selects randomly (with uniform distribution) one slot to perform its identification. Correctly identified tags withdraw from the process, while colliding ones contend again in the next frame.

Efficiency of FSA is commonly studied as the mean number of slots required for identifying a population of n tags (or conversely, as the identification throughput, i.e. identifications per slot). The usual goal is to minimize the mean reading time (or maximizing throughput) by establishing some suitable frame-length selection policy.

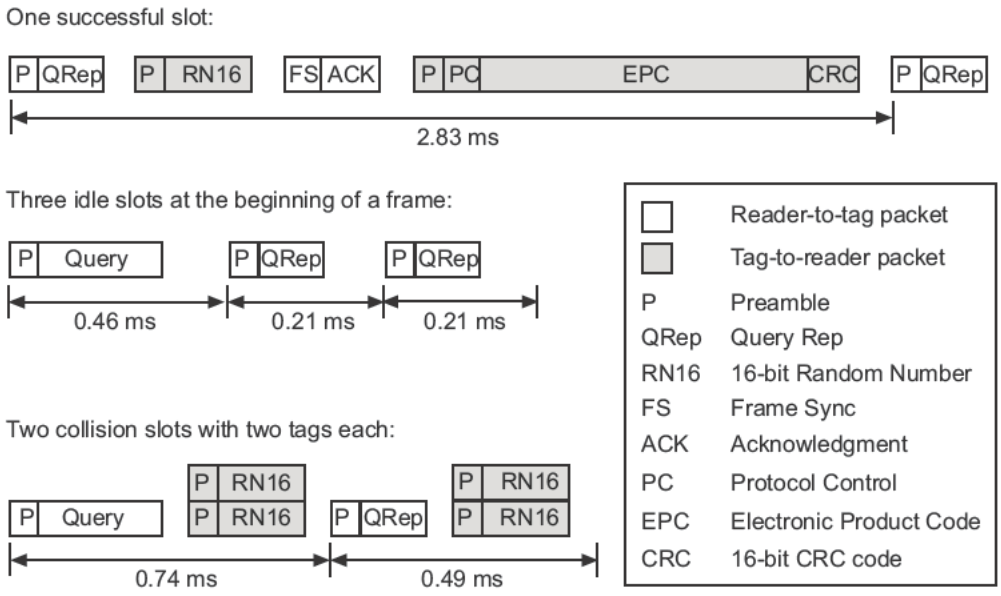


Figure 3.1: Durations of different command sequences in an ISO 18000-6C frame computed from the timing parameters specified in [Floerkemeier and Sarma, 2009]

Some examples are works [Lee et al., 2005; Zhu and Yum, 2010; Vales-Alonso et al., 2011], which study DFSA proposals where efficiency is measured at a slot-level. A well-known DFSA approach is maximizing throughput per frame by setting k to the estimated number of contenders $\hat{n}$. Assuming EPC constraints, in [Vales-Alonso et al., 2011] the best frame-length assignment to maximize throughput per frame is provided. As in previous chapters, DFSA will denote this criterion, assuming perfect tag count estimation, i.e. $\hat{n} = n$.

However, considering efficiency in terms of slots is too simplistic to study real RFID systems, since slot durations vary considerably in practice. Figure 3.1 shows the duration of different command sequences corresponding to successful, empty slots and collision (respectively t_S, t_E, and t_C). The detailed configuration of the timing parameters is provided in [Floerkemeier and Sarma, 2009]. Note that different configurations would result in different command durations. In particular, since the number of successful identifications (S), empty slots (E) and collisions (C) are random, a frame has also a random duration.

In addition, the dynamic evolution of the identification process affects the efficiency of a FSA system as well. As the reading process takes place, the number of unidentified tags decreases and the frame-length may vary from frame to frame. Thus, both effects must be considered to measure accurately the efficiency of FSA. Most works use simulators to take these effects into account either measuring efficiency in terms of slots (e.g. [Lee et al., 2005; Vales-Alonso et al., 2011; Chen, 2009]) or considering time efficiency. For example, [Khandelwal et al., 2007] obtain by means of simulations a mean reading time per tag of 0.58 ms for a non-standard

configuration (authors assume t_S=t_C) and using a DFSA mechanism with ideal tag count estimation.

On the other hand, some authors have developed analytical dynamic models to compute efficiency mostly in terms of slots (or even in terms of frames). [Vogt, 2002] models the RFID identification process by means of a Discrete Time Markov Chain (DTMC) evolving frame-by-frame (with frames of static length k). This DTMC allows computing the minimal number of *frames* required to identify a population of n tags with a confidence level α. Besides, Vogt obtains also the identification time (in seconds) by approximating the frame duration by an average obtained experimentally. [Bueno-Delgado and Vales-Alonso, 2011] use a DTMC to compute the mean identification time (in slots) to identify n tags considering SFSA and DFSA. Moreover, an approximation to the mean identification time (in seconds) is provided based on the (inaccurate) assumption that identifications and collisions are independent in a frame. [Zhu and Yum, 2010] also formulate a DTMC, but with the goal of computing the best policy for frame-length adjustment in DFSA in order to maximize slot-level throughput.

Contrary to previous works, we compute exactly the *mean number of identifications* within a bounded time t when a population of n tags contend in FSA, considering the dynamic identification evolution. It is fundamental to remark that previous works evaluate final throughput (ρ identifications per second). That is, compute ρ=$n/\overline{t(n)}$, where $\overline{t(n)}$ denotes the mean identification time of the population of n tags. Therefore, the mean number of identifications within a time t is usually assumed to be given by ρt. However, this is false since the throughput ρ is not constant during the identification process as will be demonstrated in Section 3.4. Besides providing exact results, the method developed in this work has the advantage that it can be applied to any given frame-length selection policy. As an example, we consider SFSA and DFSA with the settings and frame-length constraints of the EPC standard. The main utility of the result is solving the question: How many tags will be identified in a given time? A question that remained unsolved analytically.

Beyond obtaining FSA performance, results allow also to obtain the best frame-length configuration for SFSA systems with bounded identification time. Another direct application is formulating schedulers in RFID environments with multiple readers (see [Eom et al., 2009; Ferrero et al., 2012]). Due to mutual interferences, readers must operate multiplexed in time, allocating to each reader an exclusive time period, such that the total available time is divided among readers. Most works formulate schedulers heuristically (e.g. [Eom et al., 2009; Ferrero et al., 2012]). However, from our results, the expected number of identifications within each period can be predicted allowing to select the best configurations to maximize global performance.

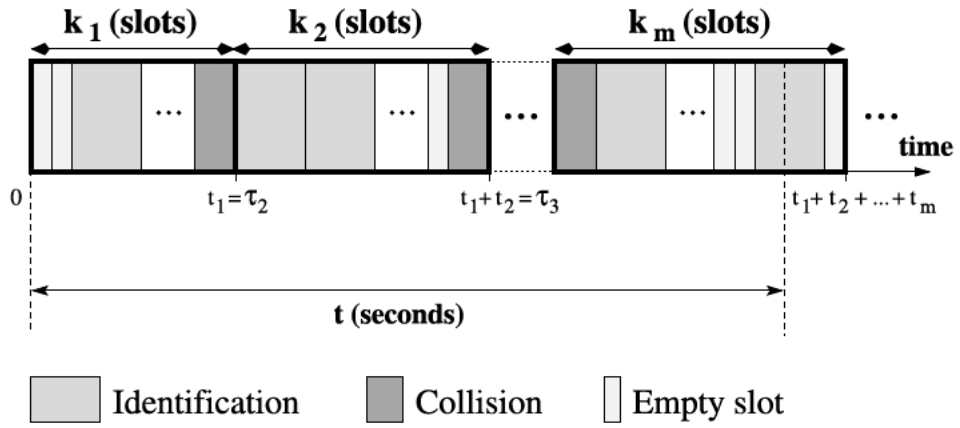


Figure 3.2: Reading process model

In next section the model of the system is introduced formally. Section 3.3 develops the analytical method to compute the mean identification time. Section 3.4 discusses main results, and Section 3.5 concludes this work.

3.2. System model

Fig. 3.2 summarizes the main concepts of our model. We assume that n tags begin the identification process at time 0, and new tags are not added during the process. Then, a number of interrogation frames i=1,2,3,... are consecutively allocated until eventually all tags are identified. The frame-length is denoted by k_i for the frame i. Besides, n_i will denote the number of tags not identified at the beginning of frame i. The duration (in seconds) of frame i is denoted by t_i, and is a random variable since it depends on the number of identifications (S_i), collisions (C_i), and empty slots (E_i), which are random. Let us remark that these variables are related by the condition E_i=$k_i - S_i - C_i$. Thus, the bivariate random variable (S_i, C_i) fully specifies the event that occurs at frame i. Finally, let τ_i denote the time at which frame i begins. That is, τ_i=$t_1 + \ldots + t_{i-1}$ and it is obviously random.

The goal is, given a time period t, computing the mean number of identifications during such period, which is denoted $\varphi(n,t)$ henceforth. As shown in Figure 3.2 the frame end might not be aligned with this time t in a general case. Thus, the final frame m might be partially allocated within the time limit t. Only tags identified before t are considered in the count. Note that this condition implies that if a slot with successful identification is partially in t interval (as in the example of Figure 3.2), this identification must not be counted. In addition, note that the time remaining for the identification at beginning of frame i is $t - \tau_i$.

In FSA, it is possible that the first slot in a frame has a longer duration than successive slots, since *Query* packets are longer than *Query Rep* packets (as shown in Figure 3.1). However, in our analysis this characteristic is not introduced since

the effect is negligible and adding it will entangle notation, but it would be straightforward to add this aspect to the analytical method discussed next.

3.3. Identification performance analysis

Given the frame i of length k_i where n_i tags contend, the resulting random selection of slots by the tags leads to a particular random event of identifications and collisions (S_i, C_i). The duration of the frame (in seconds) is given by:

$$t_i = S_i \times t_S + C_i \times t_C + (k_i - S_i - C_i) \times t_E \tag{3.1}$$

The mean number of identifications from frame i onwards is given by $\varphi(n_i, t-\tau_i)$. If we assume that frame i lies entirely within the identification period t, then it can be decomposed as:

$$\varphi(n_i, t - \tau_i) = \mathbb{E}_{(S_i,C_i)}\{S_i + \varphi(n_{i+1}, t - \tau_{i+1})\} \tag{3.2}$$

That is, the expected number of identifications in frame i, S_i (see [1]), plus the number of tags identified from frame $i+1$ onwards, $\varphi(n_{i+1}, t - \tau_{i+1})$. Since n_{i+1} $= n_i - S_i$ and $t - \tau_{i+1}$=$t - \tau_i - t_i$, the latter part of Eq. (3.2) can be rewritten as $\varphi(n_i - S_i, t - \tau_i - t_i)$, leading to:

$$\varphi(n_i, t - \tau_i) = \mathbb{E}_{(S_i,C_i)}\{S_i + \varphi(n_i - S_i, t - \tau_i - t_i)\} \tag{3.3}$$

Although from the perspective of frame $i+1$ the expression $\varphi(n_i - S_i, t - \tau_i - t_i)$ is a mean, it is actually a random variable from the perspective of frame i, since it depends on S_i and C_i (see [2]). Therefore Eq. (3.3) coherently includes $\varphi(n_i - S_i, t - \tau_i - t_i)$ inside the expectation operator.

Next, given the joint probability of the number of successful identifications and collisions in a FSA frame with k_i slots and n_i contenders, $P[S_i = s, C_i = c|k_i, n_i]$, the expectation given by Eq. (3.3) can be expanded, leading to Eq. (3.4).

$$\varphi(n_i, t - \tau_i) = \sum_{s=0}^{\min\{n_i,k_i\}} \sum_{c=0}^{\min\{\left\lfloor \frac{n_i - S_i}{2} \right\rfloor, k_i - S_i\}} s + \varphi(n_i - s, t - \tau_{i+1}) P[S_i = s, C_i = c|k_i, n_i] \tag{3.4}$$

In this expression the summation limits reflect the possible configurations of (S_i, C_i). Namely, the number of successful identifications is limited both by the number of contenders n_i and the slots available k_i. On the other hand, the number

[1] Note that if frame i is partly outside period t the number of identifications within t would be less than S_i.

[2] As shown in Eq. (3.1) t_i depends also on k_i, but k_i is deterministic from frame i perspective.

$$P[S_i = s, C_i = c|k_i, n_i] =$$

$$= \frac{k_i! n_i!}{(k_i - s - c)! s! k_i^{n_i}} \sum_{a=0}^{\min\{n_i - s, c\}} \sum_{b=0}^{c-a} (-1)^{a+b} \frac{1}{a! b! (n_i - a - s)!} \frac{(c - a - b)^{n_i - a - s}}{(c - a - b)!} \quad (3.5)$$

$$\varphi(n_i, t - \tau_i) = \sum_{s=0}^{\min\{n_i, k_i\}} \sum_{c=0}^{\min\{\left\lfloor \frac{n_i - S_i}{2} \right\rfloor, k_i - S_i\}}$$

$$\begin{cases} s + \varphi(n_i - s, t - \tau_i - t_i) P(S_i = s, C_i = c|n_i, k_i) & \text{, if } t - \tau_i > t_i \\ \psi(t - \tau_i, k_i, s, c) P(S_i = s, C_i = c|n_i, k_i) & \text{, otherwise} \end{cases} \quad (3.6)$$

$$\psi(t - \tau_i, k_i, s, c) = \begin{cases} 0 & \text{, if } k_i \leq 0, \text{ or } t - \tau_i \leq t_S \\ (1 + \psi(t - \tau_i - t_S, k_i - 1, s - 1, c)) \frac{s}{k_i} + & \\ + \psi(t - \tau_i - t_C, k_i - 1, s, c - 1) \frac{c}{k_i} + & \\ + \psi(t - \tau_i - t_E, k_i - 1, s, c) \frac{k_i - s - c}{k_i} & \text{, otherwise} \end{cases} \quad (3.7)$$

of collisions is limited by the maximal number of groups with two tags that can be formed with $n_i - S_i$ contenders, $\lfloor (n_i - S_i)/2 \rfloor$, and the available slots after the successful identifications $k_i - S_i$.

Besides, [Chung et al., 2013] have obtained analytically the joint probability $P[S_i = s, C_i = c|k_i, n_i]$, which is given by Eq. (3.5) using our notation, and allows direct computation of Eq. (3.4).

Nevertheless, in the general case, it is possible that the considered frame i lies partly outside the t interval. It occurs when $t_i > t - \tau_i$. In this case, only the slots with valid identifications inside T must be computed. Let us denote by $\psi(t - \tau_i, k_i, S_i, C_i)$ the expected number of identifications within $t - \tau_i$ seconds in a frame of k_i slots with S_i identifications and C_i collisions when $t_i > t - \tau_i$. Then Eq. (3.4) can be modified to take into account that frame i can lie outside period t. Eq. (3.6) shows its final expression.

Then, $\psi(\cdot)$ must be computed. Although (S_i, C_i) is fixed to a value (s, c), the particular order of the slots (success, collision or empty) is random. The probability that first slot corresponds to an identification is $\frac{s}{k_i}$ since tags select slots with a random uniform distribution. Equivalently, the probability of the first slot having collision is $\frac{c}{k_i}$, and being empty $\frac{k_i - s - c}{k_i}$. The count of identifications has to be increased by 1 only in the first case. This same reasoning can be applied to successive slots considering that frame-length is decreased by one after each slot, and that the count of remaining identifications or collisions is decreased if that event

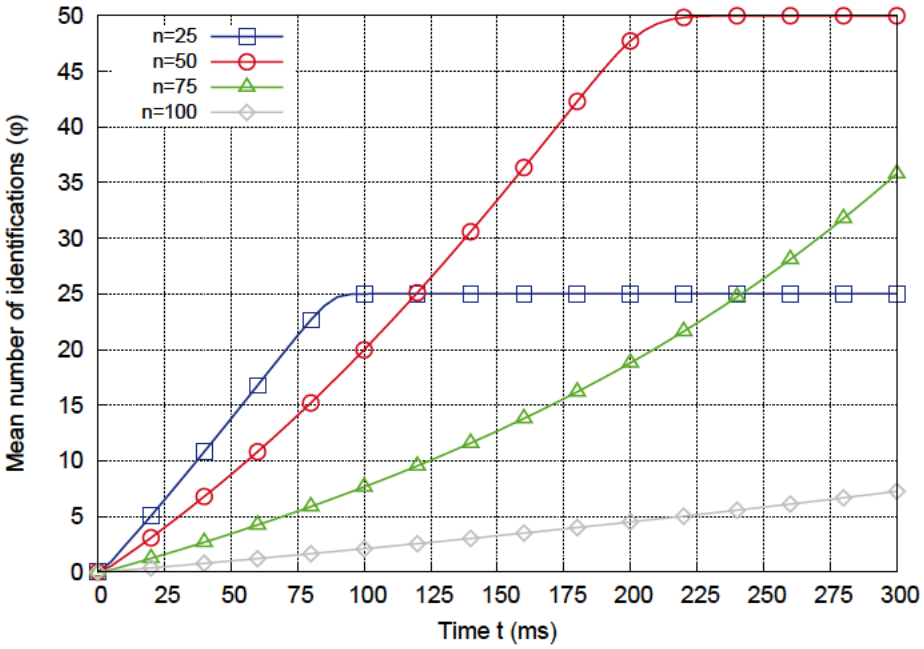


Figure 3.3: Mean number of identifications versus time, SFSA k=16

occurs. Hence, $\psi(\cdot)$ can be expressed recursively through Eq. (3.7).

Note that the final Eq. (3.6) is independent of the frame i considered. In particular, considering frame 1, $\varphi(n_1, t-\tau_1)$ leads to computation of $\varphi(n,t)$, since n_1=n and τ_1=0. This expression can be evaluated recursively since the departure variables n or t are both deterministic. Finally, let us remark that the particular frame-length adjusting mechanism must be known to evaluate k_i.

3.4. Results

Following the procedure of the previous section we have computed analytically the mean identification time for different FSA configurations. The slot durations were set to t_S=2.83 ms, t_C=0.49 ms, and t_E=0.21 ms, according to Figure 3.1. Besides, all experiments have been verified by Monte Carlo numerical simulation, being totally coincident with the analysis [3].

Figure 3.3 and Figure 3.4 show the mean number of identifications versus t for different initial values of n. There are several notable aspects highlighted by these results. For SFSA, the mean number of identifications depends strongly on N, e.g. in Figure 3.3 for t = 100 ms, $\varphi(25, 100\text{ ms})$ is nearly 25 tags but it drops below 5 tags for $\varphi(100, 100\text{ ms})$. Indeed, for SFSA $\varphi(n,t)$ grows non-linearly with time, e.g. in Figure 3.3 $\varphi(75, 125\text{ ms})$=10.04 but $\varphi(75, 250\text{ ms})$=26.37, instead of 20.08.

[3] In all experiments the analytical result is within a confidence interval of ±1% of the numerical average with a confidence level at least of 99%.

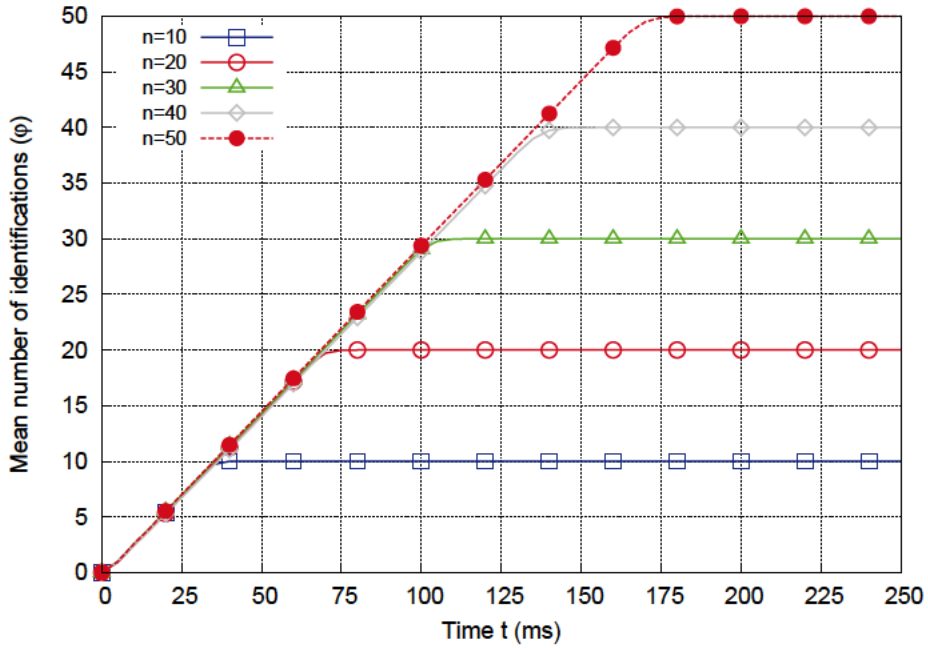


Figure 3.4: Mean number of identifications versus time, DFSA

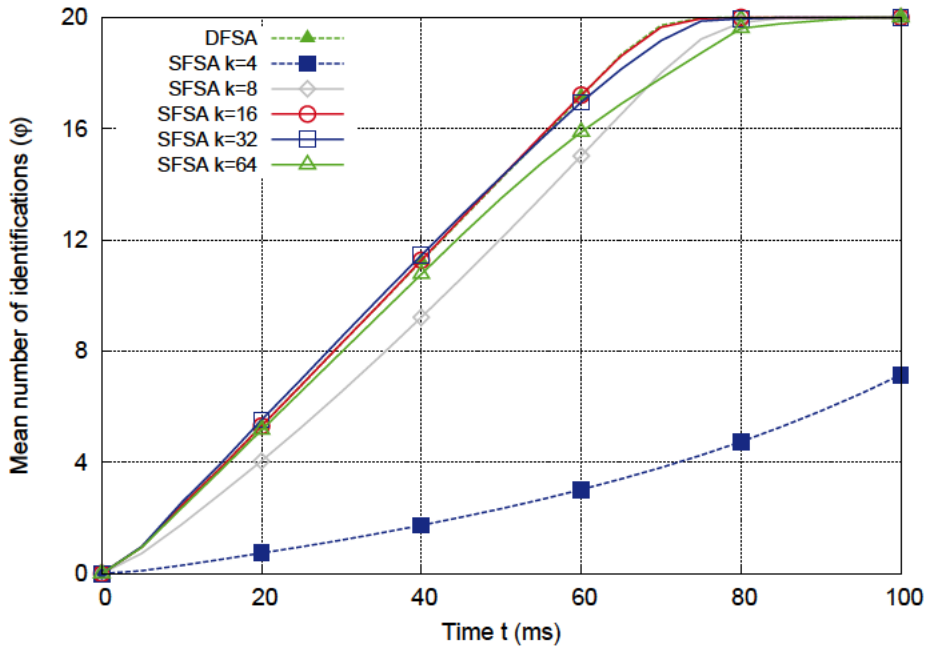


Figure 3.5: Mean number of identifications versus time, n=20

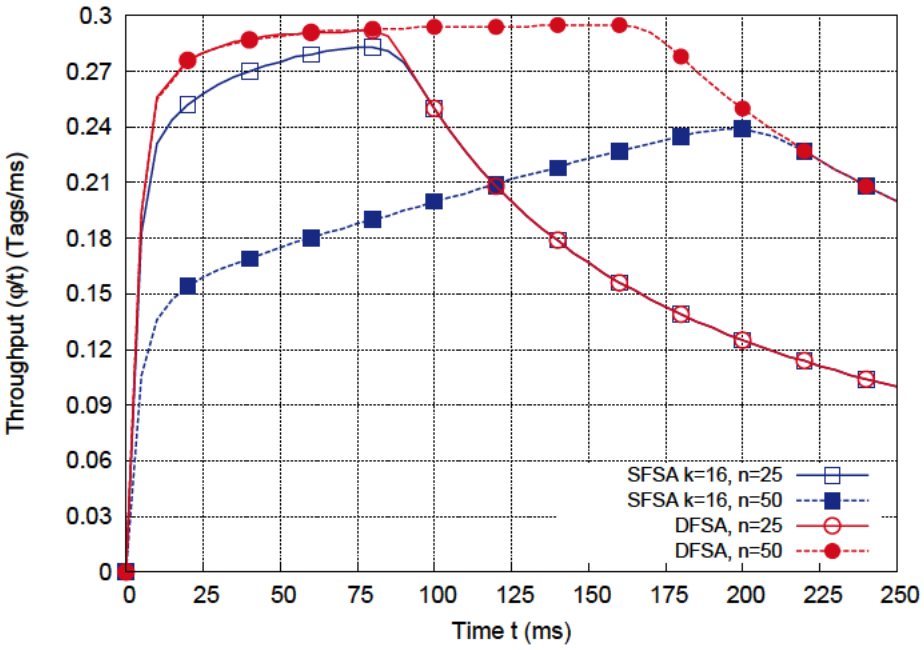


Figure 3.6: Throughput versus time

However, for DFSA the tendency is almost linear until all tags have been identified as shown in Figure 3.4.

Figure 3.6 shows the evolution of throughput, computed as $\rho=\frac{\varphi(n,t)}{t}$, for N=25 and 50. In all cases it increases until all tags are identified, and then decreases with time. In the growing regime, ρ is clearly non-constant, but for DFSA it tends to stabilize, hence the linearity shown in Figure 3.4. In all the tests performed with DFSA, identification time was between 3.3 ms and 3.7 ms per tag in this stable operation zone, with a minor influence of the initial population N. This result contrasts remarkably with the figure of 0.58 ms given in [Khandelwal et al., 2007], or the figure of 100 tags read in 460 ms used in some works dealing with RFID schedulers (e.g. [Eom et al., 2008; Ferrero et al., 2012]).

Finally, Figure 3.5 shows the effect of the frame-length selection policy, departing from the same population n=20. While for most combinations of n and t DFSA achieves the best performance, surprisingly in the example of Figure 3.5 for times $t < 51$ ms SFSA k=32 performs slightly better. Let us remark that DFSA aims at maximizing the identifications per slot at each frame, but this does not guarantee maximal throughput (in terms of identifications per second), as can be checked in the previous example. In addition, for times $t > 51$ ms, SFSA k=16 has a better performance than SFSA k=32. This observation is of practical interest since most commercial RFID readers operate only in SFSA modes. While core works of RFID aim at selecting k based only on the population n, as discussed t would also affect frame-length selection in systems with a bounded time available for readings. Fig-

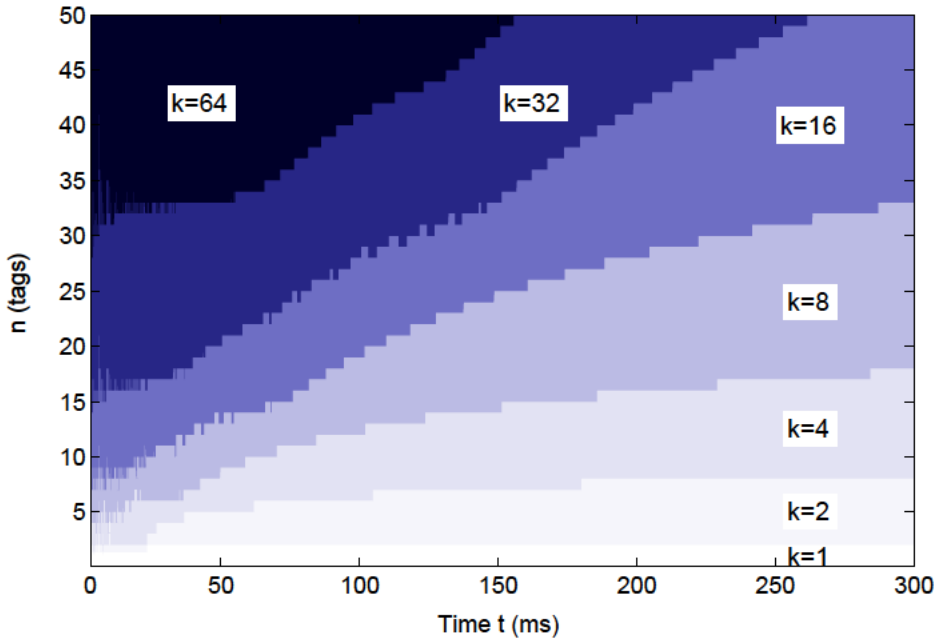


Figure 3.7: SFSA optimal frame-length configuration versus n and t

ure 3.7 shows the optimal SFSA frame-length that should be selected in this case as a function of n and t.

3.5. Conclusions

We have determined analytically the mean number of identifications from a population n of tags within a bounded time interval t. This method can be easily adapted to any frame-length selection mechanism, and allows characterizing FSA systems. As demonstrated in Section 3.4, readings depend on the initial population n and there are non-linear effects which have been neglected before in RFID works. We hope that the results of this work may help to improve different applications, such as RFID schedulers, tag count estimators, or developing new DFSA policies.

Part II

Wireless Sensor Networks Deployment

– Why did they pick you?
Because you like to fight?
– I'm expendable.
– What mean expendable?
– It's like someone invites you to
a party and you don't show up.
It doesn't really matter.

Wireless Sensor Networks (WSNs) offer new capabilities to study large-scale physical world phenomena, leading to unprecedented observation abilities. Despite the continuous development of WSNs, there still exists a number of open research challenges. Network deployment, for example, remains a critical issue from an operational and efficiency perspective [Younis and Akkaya, 2008]. Deployment planning requires consideration of account several objectives such as energy consumption, sensing coverage, network lifetime, network connectivity, and so forth. Often these objectives are conflicting, and operational trade-offs must be established during network design.

The quality of the sensing procedure and data acquisition is crucial for the network to efficiently perform its intended task, and has to be considered at an early design stage. Sensing quality depends on the *coverage* of the network. In a wide sense, this issue comprises many aspects regarding different properties of

the network. In fact, in sensor nodes, the term "coverage" may refer to different concepts, for instance: *wireless coverage* refers to the area around a node were it can effectively communicate, whereas *sensing coverage* should be interpreted as the area around a node were it can effectively sense some physical magnitude. The former is related to the connectivity properties of the network and to a variety of associated problems, connectivity being a constraint in most of them, i.e. minimizing some parameter of interest such as energy consumption. The latter determines the quality of the sensed information in general, which is the main topic of another line of research [Meguerdichian et al., 2001].

We consider a particular class of sensing coverage problems, defined in [Poduri and Sukhatme, 2004] as *constrained coverage* problems, that is, finding a deployment configuration that maximizes the collective sensing coverage of the nodes, while satisfying one or more constraints. The main difference with previous works is that we contemplate that some zones in the target area are more relevant than others, leading to a more realistic scenario.

Exploration rover missions in Mars or the Moon help us illustrate our discussion. A Rover can be deployed in small regions for carrying out in situ measurements, but the data collected will be limited to a specific location and time. Besides, the mobility and operation of a Rover over a permanently shadowed and rough region can be extremely challenging [Prasad and Murty, 2011]. WSNs may become a feasible alternative to extend their monitoring range, by deploying clusters of sensors over the interest areas. The interest area is defined by the probability of occurrence of some event of interest. The question we face then is how to spread the sensors over those areas to maximize the probability of capturing events of interest. In addition, since nodes have only short-range communication capabilities, they must report the captured events to a special node, the *sink*. In the Martian exploration example, the sink is some special probe landed close to the sensors, which is in charge of communicating with the nodes and relaying information to Earth. Nodes can communicate via multi-hop links to reach the sink, allowing large areas to be covered. However, if a node or a group of nodes are isolated, their events will be lost. This leads us to a *constrained coverage* problem, since a connected network is compulsory.

At this point, two possibilities can be considered in order to deploy the sensing nodes:

- *Deterministic*: we assume that sink and sensor position can be individual and precisely selected. The solution to the constrained coverage problem provides the set of node locations.

- *Random*: we assume that the position of the probe can be precisely selected, but the nodes are randomly spread around it (the probe itself launches the nodes, which is called a *ballistic surface deployment*).

In Chapter 4 we focus on a specific *deterministic* case with the purpose of real-time gunshot locating, in order to detect poaching. In this problem, the coverage degree determines the accuracy of the shot location, and the goal function aims at providing the maximal accuracy, given a maximal network cost. We propose a mathematical model whose optimization methodology includes a combination of a derivative-free descent method with a Pareto frontier that represents the equilibrium between network cost and coverage.

In Chapters 5 and 6 we address two optimization *random* deployment problems, where nodes are spread following a Gaussian random distribution. An optimization problem which maximizes network sensing coverage is proposed in Chapter 5, where we assume that the deployment takes place over a plain surface and nodes are grouped in several clusters (given a predetermined number of both nodes and clusters). The goal of this model is to decide the launch point and dispersion for each cluster in order to maximize the 1-degree coverage of the most relevant areas. This corresponds to real situations where clusters are dropped in an airborne launch in which dispersion is controlled by the release altitude, and where redundancy in the coverage is not necessary.

In Chapter 6 we introduce a thorough analytical description of deployments of WSN for in-situ Lunar exploration missions. This type of deployments are subject to a random scattering of sensors as well, and results in non-deterministic observation capabilities. Proper models reflecting this property as well as the energy and mass constraints imposed by a real mission are presented. From them, an optimization problem which allows to search optimal deployment planning variables in order to maximize the observational capabilities of the network is defined, considering as in the previous problem the goal of maximizing the 1-degree coverage of most relevant areas.

Chapter 4

Acustic Sensor Planning for Gunshot Location in National Parks: A Pareto Front Approach

Man fears the darkness, and so he scrapes
away at the edges of it with fire.

4.1. Introduction

The Spanish SEPRONA [Guardia Civil, Spanish Ministry of Interior, 2009] agency fights against poaching with considerable success. However, the problem remains relevant. Since 2007, SEPRONA has detained over 150 people for hunting felonies [ELPAÍS.com, 2008; News HOY.es, 2009]. Consequently, we have proposed the DiANa project, *Detección de caza furtiva con Armas de fuego en parques NAcionales* (Detection of illegal hunting with gunfires in national parks), to automatically detect and locate gunshots, which is endorsed by Cabañeros National Park [Spanish Goverment, 2004].

The DiANa system consists of a network of acoustic sensors that locate gunshots in wide open area extensions. The system consists of a network of acoustic sensors that locate gunshots by hyperbolic multi-lateration estimation. The differences in sound time arrivals allow to compute a low error estimator of gunshot location. Among the methods of detection we have utilized the time difference of arrival (TDoA) methodology. The TDoA technique exploits the relationship between distance and transmission time when the propagation speed is known. Once the time delays are calculated, they are processed in order to estimate the location of the source. Due to the distances between the deployed sensors (in the order of hundreds

Figure 4.1: Landscape of the Cabañeros National Park

of meters), and the smooth landscape in Cabañeros National Park (Figure 4.1), we have assumed a two-dimensional scenario.

TDoA-based ranging techniques require accurate clock synchronization. Every sensor knows its own position exactly, and it records the arrival time of the sound event. So, the sensor clocks must be as tightly synchronized as possible, using dedicated time synchronization algorithms. Since an acoustic signal location in open spaces network must scale to large sizes, it is necessary to minimize the number of exchanged messages to attain convergence, keeping energy consumption at reasonable levels if the electric grid is not available.

Selecting a synchronization schema for real applications like ours is not easy. The best known synchronization schemas implement network mechanisms to adjust all local clocks to the same value. This is achieved by exchanging time stamps between node pairs. The more frequent the exchanges, the higher the time accuracy. Two representative examples are Reference Broadcast Synchronization (RBS) [Elson et al., 2002] and the Timing-Sync Protocol for sensor Networks (TSPN) [Ganeriwal et al., 2003]. We discarded GPS receivers for their high cost. The Section 4.2.3 present a new ad-hoc flood method to set the clock times in every node in the network to the same value. In this method, the nodes do not exchange synchronization messages, and thus they save power. Once a node detects, for example a gunshot, the time of the event is transmitted to the sink node through a previously generated path. The method performs a cooperative backward time adjustment, so that every node along the path to the sink is able to estimate the event time of the node that receives the time stamp.

Regarding network planning, since the areas under surveillance are wide, and electric power is seldom available in them, it is necessary to maximize detection

coverage and minimize system cost at a time. Therefore, we model sensor network planning as an unconstrained problem with two objective functions. We provide a set of candidate solutions of importance by combining a derivative-free descent method and a Pareto front approach.

Due to the inherent difficulties exhibited by the thus far formulated models in sensor network planning, several heuristic optimization strategies have been proposed in the literature: variants of simulated annealing [Anderson and McGeehan, 1994], genetic algorithms [Lieska et al., 1998], gradient descent (when applicable) [Sherali et al., 1996] and others [Kamenetsky and Unbehaun, 2002; Unbehaun and Kamenetsky, 2003].

Some of these approaches (simulated annealing, genetic algorithms and the like) do not guarantee theoretical convergence. Regarding gradient descent methods, the gradient is often unavailable or too costly to compute. Therefore, we have adapted a non-monotone derivative-free optimization technique with guaranteed convergence [García-Palomares et al., 2006] to formulate and solve a computationally efficient **optimization model** with a dual objective: maximizing acoustic network coverage and minimizing power infrastructure cost. The results are presented as a Pareto front, revealing solutions that are clearly superior to random seeding.

Our notation is as follows: Lower case Greek letters are scalars, lower case Latin letters are vectors in $\mathbb{R}^2, x_k^j$ is the jth component of the vector x_k, and $||x||$ is the Euclidean norm. A capital Latin letter, say S, stands for a collection of vectors in $\mathbb{R}^2$ if $S = \{s_1, \ldots, s_p\}$ we also say that $S \in \mathbb{R}^{2p}$; $\tau S = \{\tau s_1, \ldots, \tau s_p\}$, and the sum $Z = S + D$ means that S and D have the same number of elements, say p, and $z_k \in Z$ if *and only if* $\exists k \in \{1, \ldots, p\} \mid z_k = s_k + d_k$. In general the subindex i is the value of an entity (scalar, vector, set, and so forth) at the ith iteration of an algorithm; for instance $S_i = \{s_{i1}, \ldots, s_{ip}\}$ is a set of p vectors in $\mathbb{R}^2$, at the ith iteration.

4.2. Gunshot location

In this section we describe the location procedure for acoustic events, which we have implemented for MicaZ motes. The goal of our system is gunshot location by means of a sensor network. Location is based on hyperbolic positioning [Patwari, 2005]. Hyperbolic positioning requires the sensor clocks to be synchronized, in order to apply the Time Difference of Arrival (TDoA) technique. For this reason we have implemented a synchronization protocol in the MICAz motes. Next, we describe the system architecture, the location method and, finally, the synchronization schema.

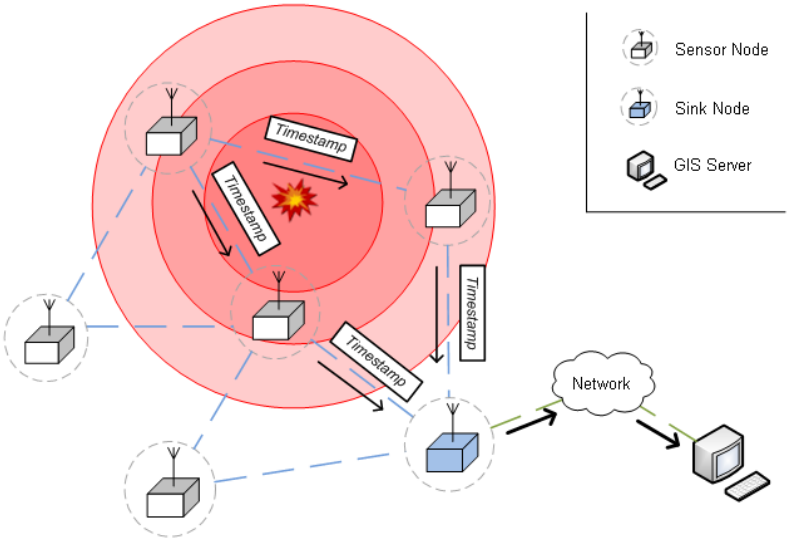


Figure 4.2: Gunshot location architecture

4.2.1. System architecture

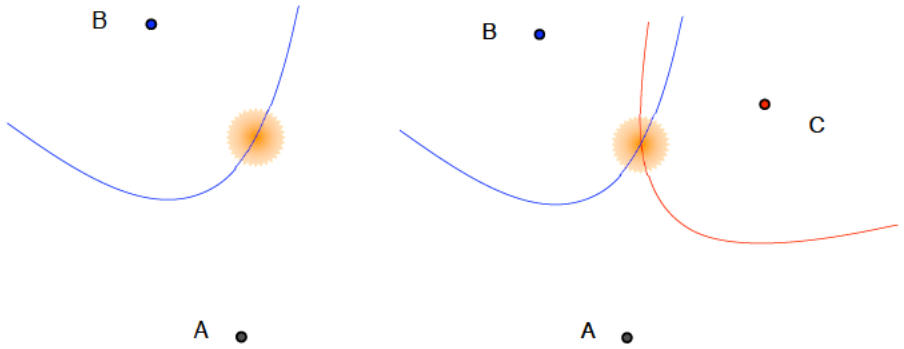
Figure 4.2 shows the system architecture, with three components:

- Sensor nodes: The sensor nodes in known positions are equipped with the necessary hardware for detection of acoustic events. They can discriminate between *normal* and *shot* segment classes in audio streams. When a sensor node detects a sound event, it transmits a packet with information about the type of sound event and a sound time-stamp to a special node, the sink.
- Sink nodes: Sink nodes collect the packets sent by sensor nodes and deliver them to the GIS server to calculate the position of the sound event. Sink nodes may be sensing nodes as well.
- GIS server: Using the information of the sink nodes, the GIS server estimates the position of the acoustic event by means of a hyperbolic method, described next.

4.2.2. Acoustic source location procedure

Since our optimization model (see Section 4.3) considers a flat landscape with scattered trees, we have selected a two-dimensional hyperbolic positioning algorithm. Nevertheless, it could be easily extended to three-dimensional location in rough scenarios.

Let us consider an acoustic event that takes place at an unknown position $x \in \mathbb{R}^2$, which we wish to determine. Formally, all sensing node locations $s = (s^1, s^2)$ belong to a well defined *compact* set $X \subset \mathbb{R}^2$. We denote the *Euclidean* distance



(a) Locus of the TDoA obtained between positions x_A and x_B.

(b) Locus of the TDoA obtained between positions x_A and x_B, and between positions x_A and x_C.

Figure 4.3: Hyperbolic multilateration of an acoustic signal

$\delta(s, x)$ between a sensor location s and another point x in the scenario, $s, x \in X$, by

$$\delta^2(s, x) = \|s - x\|^2. \tag{4.1}$$

Let us also assume that a subset $B \subset X$ of the nodes have detected the event, and that the position of those nodes is known. Let us denote the sound propagation speed as v. Then, the time of arrival t_b of the event at any node $b \in B$ is:

$$t_b = \frac{1}{v}\delta(x, s_b) \tag{4.2}$$

and the difference between the arrival times of the event to a pair of nodes s_b and $s_{b'}$ is:

$$\tau_{b-b'} \triangleq t_b - t_{b'} = \frac{1}{v}[\delta(x, s_b) - \delta(x, s_{b'})], \quad \forall b, b' \in B \tag{4.3}$$

This expression defines hyperboles, whose intersection determines the source of the shot. Figure 4.3 shows this process graphically.

For two given receiver locations, s_b and $s_{b'}$, a set of emitter locations would yield the same TDoA measurement. This is the *locus* of possible emitter locations and it describes a hyperbole. If we now consider a receiver at a third location $s_{b''}$, it provides a second TDoA measurement, and, hence, it allows to locate the emitter on a second hyperbole. In general, a set of distances from the source to every sensor pair identifies a set of hyperboles. As a consequence, the location of the source is the intersection of this set of hyperboles. In Section 4.5, our tests show that location accuracy increases with the number of gunshot detection nodes. In most trials,

the accuracy in scenarios with three detection nodes or more is sufficient to find the acoustic source (the detailed TDoA location technique is described in [Patwari, 2005]).

Although non-linear expression (4.3) has a unique solution if there are enough hyperboles, there is some uncertainty in the calculation since:

- The speed of sound varies depending on altitude, humidity and air temperature. As we have mentioned, multi-path propagation affects the accuracy of acoustic signal detection. Single spread-spectrum techniques such as those in [Girod and Estrin, 2001] largely mitigate it.

- The microphone directionality or polar pattern affects the result.

- The clock drift may drastically vary in time due to environmental temperature and humidity changes. In Section 4.2.3 we propose an approach to reduce sensor clock deviations.

Due to these inaccuracies, expression (4.3) may be inconsistent. Nevertheless, low-error estimations are possible. A nonlinear optimization problem can be formulated [Zhang et al., 2008] to minimize the difference between estimated and real positions. We minimize the square error of the location, defined as the differences between the squares of the theoretical and measured differential arrival times to a reference node $s_{b'}$:

$$x = \arg\min_{x} \{ \sum_{\forall b \in B \setminus s_{b'}} [(\delta(x, s_b) - \delta(x, s_{b'})) - \upsilon \tau_{b-b'}]^2 \} \tag{4.4}$$

The least square minimization problem in (4.4) is not convex. Thus, standard optimization algorithms, like incremental gradient, are not guaranteed to converge to the global minimum. The initial conditions in an iterative algorithm may lead to a local optimum or a saddle point at the termination, adding imprecision to gunshot location. The required computing power is moderate, and the GIS server (Figure 4.2) can handle the calculations. The solutions can be obtained practically in real-time (see Section 4.5). Our synchronization algorithm (see Section 4.2.3) also contributes to the location procedure, since it is required for the sink node to compute the TDoA between the detectors. The tests with our reference implementation in MICAz motes reveal that, the more sensors that detect the sound event, the greater the location precision (see Section 4.5). This is due to error compensation in arrival time measures. In conclusion, a high density of sensors around a sound event always improves location precision. Therefore, optimal sensor positioning improves the performance of the location system.

4.2.3. Synchronization schema

Time synchronization is a fundamental aspect in distributed sensor networks. In the proposed shot detection system, the differences in arrival times can only be computed if the nodes are tightly synchronized. To adjust the clocks, the nodes must exchange messages indicating the time reference. In fact, time adjustment degrades progressively due to clock drifts, and it is indeed mandatory to readjust it periodically (by sending new messages). However, power consumption is higher if synchronization packets are continuously transmitted. Even if a continuous energy source is available, it may be necessary to extend the network with autonomous nodes at its edges. Therefore, the number of exchanged messages should be as low possible, for a given accuracy goal. For gunshot location purposes, if we assume a maximum error of few meters, the maximum allowed error in time synchronization is in the order of a tenth of a second (d_e/v, where d_e is the allowed spatial error). For example, three milliseconds of clock drift will cause an estimated error of one meter, relative to the real source position. Thus, fine-grained clock synchronization is mandatory. The tests with our synchronization algorithm show a time accuracy of tenths of microseconds. The implementation has been carried out with TinyOS in MICAz devices.

The synchronization algorithm has two steps:

1. *Level discovery*: This step is similar to the level discovery stage in TPSN [Ganeriwal et al., 2003]. Before the synchronization process takes place, the network has to organize itself as a hierarchical tree, beginning at a root node (in our case we choose the sink). According to the minimum number of hops to the sink, a level is assigned to each node (level 0 to the root). To compute the tree, the process starts at the root, broadcasting a level discovery packet to the nodes at level 0. The nodes that receive this packet are marked as children of the root node, and they set their level to 1. The nodes ignore further level discovery packets with greater or equal level numbers. Then, level 1 nodes broadcast their level discovery packets, and so on. Note that this process also permits to discover optimal communication paths (in number of hops) to the root, and, thus, it is valid for network routing.

2. *Synchronization*: Once the hierarchical network structure is completed, the synchronization process may start. In general, level k nodes synchronize their children (of level $k+1$).

   Besides its own local clock, a sensor node will maintain an estimation of its synchronizer node clock in the upper hierarchical level. The approximation consists of calculating the regression line of those two clocks. Previously, the level k node receives several synchronized time-stamps of level $k-1$ (see Figure 4.4), which are broadcast following the tree structure that was created at

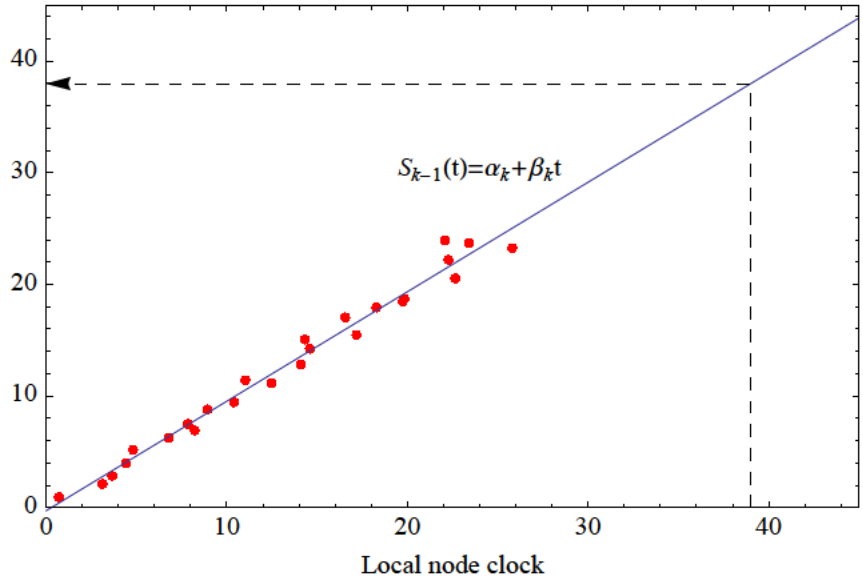


Figure 4.4: Regression line performed in a level k node, with several broadcast time stamps from a level $k-1$ node

the level discovery step. Figure 4.4 shows the regression line used to calculate the *parent* node clock in a level k node. Value α_k represents the clock offset at reference time $t = 0$, and the slope β_k is the rate of change (clock drift) of the local clock.

Once a node detects a gunshot, it sends the event to its *parent* node in the upper level, according to the parent time clock. After one or more hops, level 0 (sink node) will receive estimations of the detection time that are synchronized with the sink clock, from one or more level 1 nodes. This way, local clock exchanges do not spend power. Since clock drift varies slowly, the regression line must only be calculated every 6 or 8 hours, according to our tests with MicaZ motes. Only large temperature variations affect the regression line slope, requiring node re-synchronization.

4.3. Optimization model

The sound spectrum of a gunshot is dominated by the 130 Hz to 3 kHz frequency range [Pääkkönen, 2012]. According to [Bucur, 2006], at these frequencies there is an extra attenuation of 3 dB each ~25 m in woods, yielding 12 dB vs. 9 dB in open space. Given the sensitivity threshold of the sensing node, its maximum ranges are 1Km in open space and 750 m in a wood.

We consider a large outdoor flat scenario with open space areas and wood patches. The scenario is crossed by a few power lines. In it, we wish to deploy a given number of fixed sensing nodes, so that the detection coverage is maximum.

A point gets covered if it is reachable by a sensing node at least, although this capability increases if more sensing nodes see the point (to achieve source signal location we require a coverage of three nodes at least, as explained in Section 4.2.2).

At the same time, we wish to minimize the distance between the sensing nodes and the power lines so that the cost of the power infrastructure is minimal in case the nodes are not autonomous. The nodes communicate through the electric grid itself, so that transmission coverage is not an issue of importance.

This scenario clearly prevents an optimal educated guess, specially when only a small number of sensing nodes is available.

We define $\delta(s,x) = \omega_1(s,x) + \omega_2(s,x)$, where:

- $\omega_1(s,x)$ corresponds to propagation distance through wood space.
- $\omega_2(s,x)$ corresponds to propagation distance through open space.

Given a sensor location $s \in X$ and a point $x \in X$ we say that x is visible from sensor s if:

$$\frac{1 \text{ Km}}{750 \text{ m}}\omega_1(s,x) + \omega_2(s,x) < 1 \text{ Km} \tag{4.5}$$

Let $S = \{s_1 \ldots, s_p\}, s_k \in X, k = 1, \ldots, p$, be the positions of p acoustic sensors on X. We denote by $V(x)$ the set of all sensors $s \in S$ that are visible from the location x. We also define an arbitrary grid $G \subseteq X$. Objective function f_1, which measures acoustic sensor coverage, is defined as follows:

$$f_1(S) = \sum_{x \in G} \begin{cases} 0.5 \cdot \text{card}(V(x)) & \text{if card}(V(x)) < 3 \\ 3 + 0.01 \cdot (\text{card}(V(x)) - 3) & \text{if card}(V(x)) \geq 3 \end{cases} \tag{4.6}$$

This function penalizes grid points that see less than three sensors (a gunshot in those points cannot be located with highest precision), and gives a small bonus to grid points that see more than three sensors (the minimum number of sensors for highest precision location).

A second objective function f_2 measures the cost of the sensor deployment. As in the case of sensor coverage, there are many ways to model this. In this work we assume that the cost of a sensor unit is negligible compared to the cost of a permanent power line. As we previously said, the scenario is crossed by m power lines. We define:

$$f_2(S) = \sum_{s \in S} \min_{i \in \{1,\ldots,m\}} \delta_i(s) \tag{4.7}$$

where $\delta_i(s)$ is the Euclidean distance between s and the ith power line, *i.e.*, between s and the point in that line that is closest to s in Euclidean distance.

Our ultimate task is to place p sensors on X in such a way that the coverage f_1 on X is maximized and the cost f_2 on X is minimized. Clearly, these objectives are

contradictory. Minimizing $-f_1$ (*i.e.*, maximizing f_1) tends to spread the sensors, whereas minimizing f_2 tends to concentrate them around the power lines. For that reason, it is desirable to produce the **Pareto front** [García-Palomares et al., 2008] of these two functions, which represents a pool of *candidate* solutions. A point $x^* \in X$ belongs to the Pareto front of a set of functions in X if a further decrease of one of them is not possible without causing an increase in some of them. The methodology in [García-Palomares et al., 2008] obtains joint descent directions for all the objective functions in a set, but it requires all of them to be differentiable, and that is not the case of f_1.

In our case, since there are only two objective functions, we define the following unconstrained optimization problem:

$$\underset{s_1,\ldots,s_p}{minimize} \quad f(s_1,\ldots,s_p) = (\theta-1)f_1(s_1,\ldots,s_p) + \theta f_2(s_1,\ldots,s_p) \tag{4.8}$$

By solving problem (4.8) repeatedly, assigning random values to θ in $[0,1]$, we obtain a collection of points of the Pareto front of f_1 and f_2.

Remark 1: Note that any local minima in $-f_1$ and f_2 belong, by definition, to the Pareto front.

Next section deals with the solution of the model, including a proper choice of its parameters.

4.4. Solving the optimization model

Many optimization algorithms are iterative. Starting with a solution estimate $S_1 = \{s_{11},\ldots,s_{1p}\}$, a subsequence $\{S_i\}_{i\in I} = \{s_{i1},\ldots,s_{ip}\}_{i\in I}$ is generated that hopefully converges to the solution of the problem. As it is common in all implementations, there are several parameters (*magic numbers)* the user must set. Some of them will notably influence the performance of the algorithm, and often depend upon the structure of the objective function.

4.4.1. Alternative approaches

Two methods were suggested in [Kamenetsky and Unbehaun, 2002; Unbehaun and Kamenetsky, 2003] for access point coverage optimization, a similar problem to ours: *neighborhood search* and *simulated annealing* [Locatelli, 2000], which we compared in [González-Castaño et al., 2008]. These methods have no guaranteed convergence. On the other hand, gradient descent methods converge, but they can only we applied when the objective function is smooth, which is unusual in realistic models like ours.

4.4.2. Derivative-free unconstrained minimization

The function $f(S)$, with $\theta \in (0, 1]$, is non-smooth on X with directional derivatives everywhere defined, which is a required assumption on a recent algorithm for unconstrained minimization (ignoring $\theta = 0$ is not relevant to estimate the Pareto front, because θ can be arbitrarily close to 0). Numerical results show that the algorithm is competitive with others that try to find a good local minimum [García-Palomares and Rodríguez, 2002; García-Palomares et al., 2006]. Essentially the algorithm is an iterative process that does not force the decrease of $f(S_i)$, but imposes a controlled bound $\varphi_i \geq f(S_i)$ at every iteration. More specifically, given a stepsize $\tau_i\Delta > 0$, and a unitary direction $D \in \mathbb{R}^{2p}, D = \{d_1, \ldots, d_p\}$, one iteration of the algorithm succeeds if

$$f(S_i + \tau_i \Delta D) \leq f(S_i) + \alpha_i(\varphi_i - f(S_i)) - \nu(\tau_i\Delta), \tag{4.9}$$

where $\nu(\cdot), \varphi$ satisfy **A4 - A5** given below. The point S_i is *blocked* when the algorithm fails to satisfy (4.9) on a set of directions $\{D_1, \ldots, D_n\}, n > 2p$ that positively spans $\mathbb{R}^{2p}$. It is shown in [García-Palomares et al., 2006] that under assumptions **A1 - A5** given below, the sequence of blocked points converges to a point S^* that satisfies the zero order stationary point of $f(S)$, *i.e.*, $f'(S^*, D_k) \geq 0, k = 1, \ldots, n$, where the directional derivative is nonnegative along the given directions. In theory, if **A6** also holds, then $\nabla f(S^*) = 0$, but we are aware that **A6** is seldom fulfilled for our kind of function and we do not stress this result. The reader may read [García-Palomares and Rodríguez, 2002; García-Palomares et al., 2006] to complete the details. We reproduce [García-Palomares et al., 2006, table 1] in Table 4.1.

A1. $f(S) : \mathbb{R}^{2p} \to \mathbb{R}$ is bounded below, and $\{S_i\}_{i=1}^{\infty}$ remains in a compact set,

A2. $f(S)$ has directional derivatives $f'(S, D)$ everywhere defined:

$$\eta > 0 \Rightarrow \begin{cases} f'(S, \eta D) = \eta f'(S, D), \\ f(S + \eta D) = f(S) + \eta f'(S, D) + o(\eta) \end{cases} \tag{4.10}$$

A3. The unit directions $D_1, \ldots, D_n$ positively span R^{2p}.

A4. The function $\nu(\cdot) : \mathbb{R}_+ \to \mathbb{R}_+$ is *little-o* of τ, that is: $\lim_{\tau \downarrow 0} \nu(\tau)/\tau = 0$

A5. The reference values $\{\varphi_i\}_1^{\infty}$ are upper bounds of $f(\cdot)$, *i.e.*, $\varphi_i \geq f(S_i)$ for all i, and decrease sufficiently after a given finite number of successful iterations.

A6. $f(S)$ is strictly differentiable or locally convex at all limit points of the sequence $\{S_i\}_{i=1}^{\infty}$ generated by the algorithm.

Remark 2: The simplest case of INTERPOLATE$(S, D, \tau\Delta)$ is $Z = S + \tau\Delta D, f_Z = f(Z)$, although there exist more elaborate alternatives [García-Palomares et al.,

Parameters: $\epsilon_\tau, \mu, \tau, \varphi$	
Get S, let $f_S = f(S)$	
DO `success`$= 0$	
Choose D_k, $k = 1, \ldots, n$ that positively span $\mathbb{R}^{2p}$	
FOR $j = 1$ TO n	
$[Z\ f_Z] =$ INTERPOLATE $(S, D_k, \tau\Delta)$	Remark 2
$\alpha = \min(\tau, \alpha), \phi = f_S + \alpha(\varphi - f_S)$	Remark 3
IF $(f_Z \leq \phi - \nu(\tau\Delta))$	
`success`=`success`+1	
$S = Z, f_S = f_Z$	
ENDIF	
ENDFOR	
$\varphi = f_S$	Remark 4
IF (`success`>`max_success`)	Remark 5
$\tau = \tau + 1$	
ELSE	
$\tau = \tau - 1$	
ENDIF	
WHILE $(\tau > \epsilon_\tau)$	

Table 4.1: Non-monotone derivative-free algorithm

2006].

Remark 3: If $\alpha = 0$, we obtain the monotone version of the derivative free algorithm, which converges to the local minimum in the neighborhood of the starting point.

Remark 4: **A5** holds.

Remark 5: In order to improve convergence, [García-Palomares et al., 2006] suggested to expand τ every time a *significant* number of successes is achieved. Nevertheless, convergence is also guaranteed if τ only decreases.

Regarding conditions **A1-A5**: (1) The objective function $f(\cdot) : \mathbb{R}^{2p} \rightarrow \mathbb{R}$ as defined in (4.8) is non negative by adding a constant and $S = \{s_1, \ldots, s_p\}$ remains in the compact set X, (2) $f(\cdot)$ possesses everywhere directional derivatives if $\theta \in (0, 1]$, (3) the directions of search $D_k \in \mathbb{R}^{2p}, k = 1, \ldots, n$ are easy to generate [García-Palomares and Rodríguez, 2002], (4) the choice of $\nu(\tau\Delta)$ in Section 4.5 is $0.001(\tau\Delta)^2$, and (5) $\varphi = f(S)$ as soon as all directions in D have been explored (Table 4.1 and remark 4).

4.5. Example of deployment

Deployment target is a 10 km $\times$ 10 km flat wood scenario shown in Figure 4.5. The light areas represent open space, and the dark ones represent wood patches. The scenario is crossed by three "vertical" power lines, respectively at 1.66, 5 and 8.33 Km from the left side. This terrain representation partly encloses the difficulties in modeling Cabañeros national park.

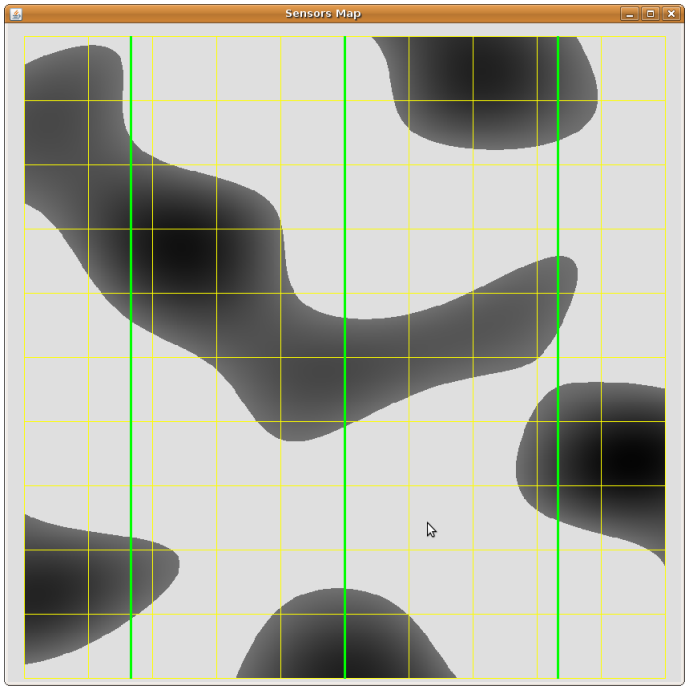


Figure 4.5: Synthetic outdoor scenario for the numerical tests. The green lines represent power lines

Grid G is given by a uniform discretization of the scenario in 50-m steps. In it we want to install 200 acoustic sensors ($p = 200$) with maximum detection coverage, at a minimum power cost. In order to obtain the Pareto front, we obtain the solution of (4.8) several times, starting from different random points S_0. We normalize the objective function by dividing it by its value at the starting points.

Since an algorithm can lead to a local optimum of (4.4) or to a solution that differs from the global optimum, we have chosen a brute force approach that computes the function in every grid point. Real-time system response is not affected, as the calculation time in a 10 km $\times$ 10 km scenario is in the order of a tenth of millisecond.

Following the results in [González-Castaño et al., 2008], instead of applying the non-monotone derivative-free search as described in Section 4.4, we applied a *zone search* variant. If we simply "move" one sensor at a time, instead of "moving" them all, the evaluation of the objective function is significantly less time consuming, since most computations in $f(S)$ do not change.

To formalize this approach, we split the scenario into q non-overlapping zones X_j, $j = 1, \ldots, q$, such that $X = \cup_{j=1..q} X_j$ and $X_k \cap X_j = \emptyset, k \neq j$. Let $V(s_i) = \{x \in G \mid s_i \in V(x)\}$. When we move the k*th* sensor, the remaining sensor positions do not change, *i.e.*, the group moves from $S = \{s_1, \ldots, s_k, \ldots, s_p\}$ to $S' = \{s_1, \ldots, s'_k, \ldots, s_p\}$. When this happens, $V(S') = V(S) \cup V(s'_k) - V(s_k)$.

To quickly obtain a *relaxed* estimate T_v of $V(s_k) \cup V(s'_k)$, we discard all $X_j \mid \forall x \in X_j \ \min(\delta(s_k, x), \delta(s'_k, x)) > 1$ Km. Let T_{nv} be the set of discarded zones and let $T_v = V(S') - T_{nv}$. Then, we compute $\sum_{x \in T_v} f(S')$, and we keep $\sum_{x \in T_{nv}} f(S') = \sum_{x \in T_{nv}} f(S)$ from the previous iterate. In the numerical tests that follow, we divided our scenario in 100 [1 Km $\times$ 1 Km] zones ($q = 100$). We compute one objective function component per zone at the beginning of the algorithm execution. When only one sensor is moved, there is no need to recompute the objective function in zones that are not affected by the sensor movement.

This way, the time to compute an objective function value drops from 20 seconds in average to just 1.5 seconds on a Pentium IV.

Let $S = \{s_1, \ldots, s_k, \ldots, s_p\}$. In order to move only one AP at a time, say s_k, we generate a set of unit search directions $d_j \in \mathbb{R}^2, j = 1, \ldots, n$ such that the set $D = \{d_1, \ldots, d_n\}$ positively spans $\mathbb{R}^2$. We recall that $n \geq 3$. We declare a *success* when

$$f(s_1, \ldots, s_{k-1}, s_k + \tau_i d_j, s_{k+1}, \ldots, s_p) \leq f(S) + \alpha_i(\varphi_i - f(S)) - \nu(\tau_i)$$

for some $d_j \in D$ and some $s \in S$. The point S is *blocked* if the algorithm is unable to move a single $s \in S$. We observe that this schema may be carried out simultaneously on a multi processor environment and it is straightforward to show that after we try all $s \in S$ we have searched on a set of directions that positively span $\mathbb{R}^{2p}$, although we are using at least $3p$ directions of search.

We tuned the method in preliminary trials and determined the following parameter values:

- $\Delta = 0.1$ Km, *i.e.*, the method stops when the maximum sensor displacement is under 100 m.
- $\epsilon_\tau = 1$
- $\tau = 5$
- `max_success`= 40 (20%p), but we do not allow $\tau > 8$.
- We initially set φ to the value of the objective function at the starting point.
- $\alpha = 0$, *i.e.*, we perform a *monotone* derivative-free search.
- Finally, we set $\nu(\tau\Delta) = 0.001(\tau\Delta)^2$.

Figures 4.6, 4.7 and 4.8 show the results. Instead of representing f_2 versus f_1, we represent f_2 versus the coverage areas (in percentage) that correspond to $V(x) \geq 1$, $V(x) \geq 2$ and $V(x) \geq 3$, respectively. Algorithm execution is repeated 20 times, from 20 different starting points. Each starting point consists of a random deployment of 200 acoustic sensors in the scenario in Figure 4.5, with a uniform distribution.

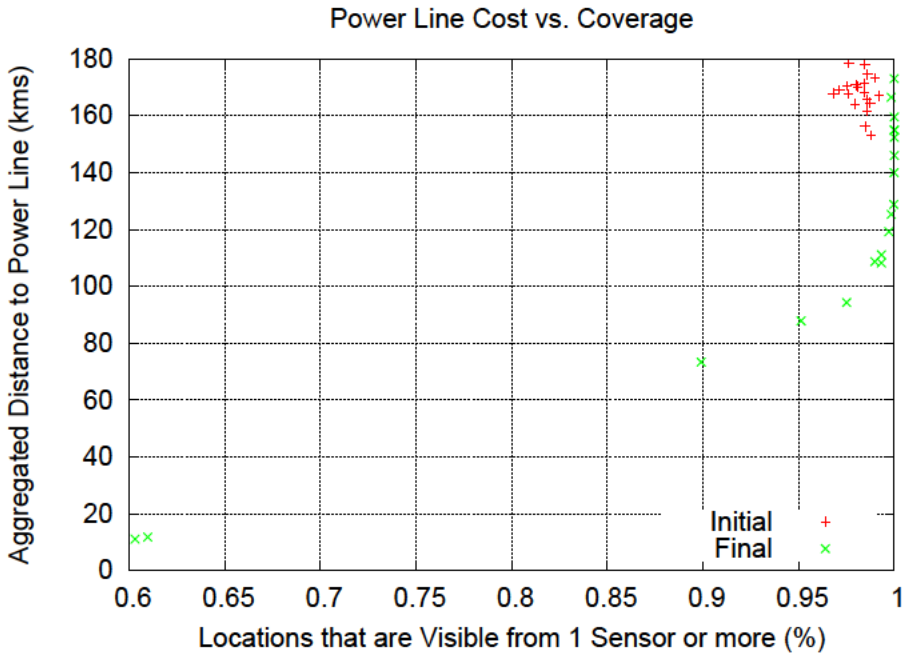


Figure 4.6: Aggregated distance to the power lines (cost) vs. coverage area with $V(x) \geq 1$

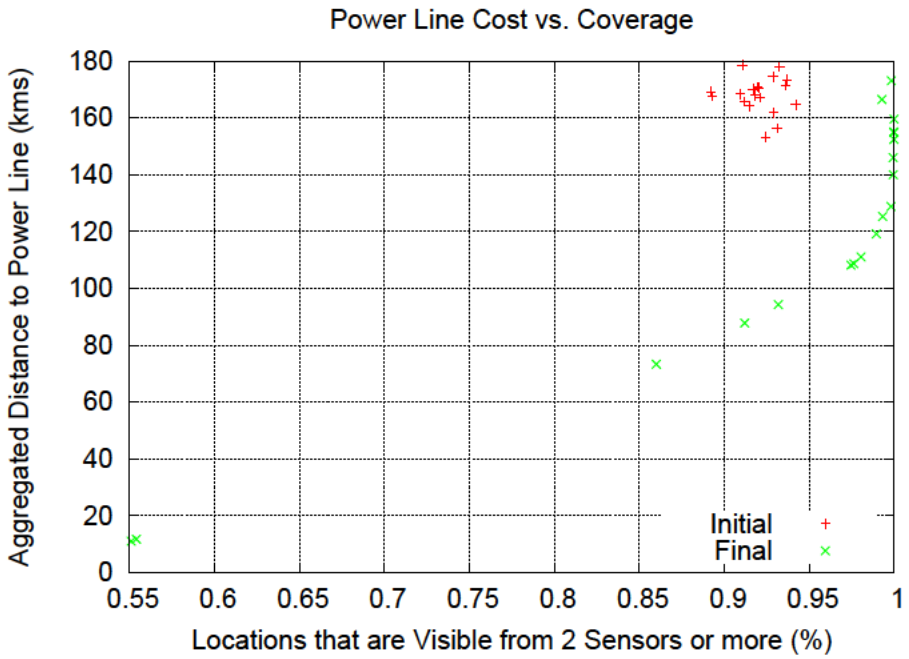


Figure 4.7: Aggregated distance to the power lines (cost) vs. coverage area with $V(x) \geq 2$

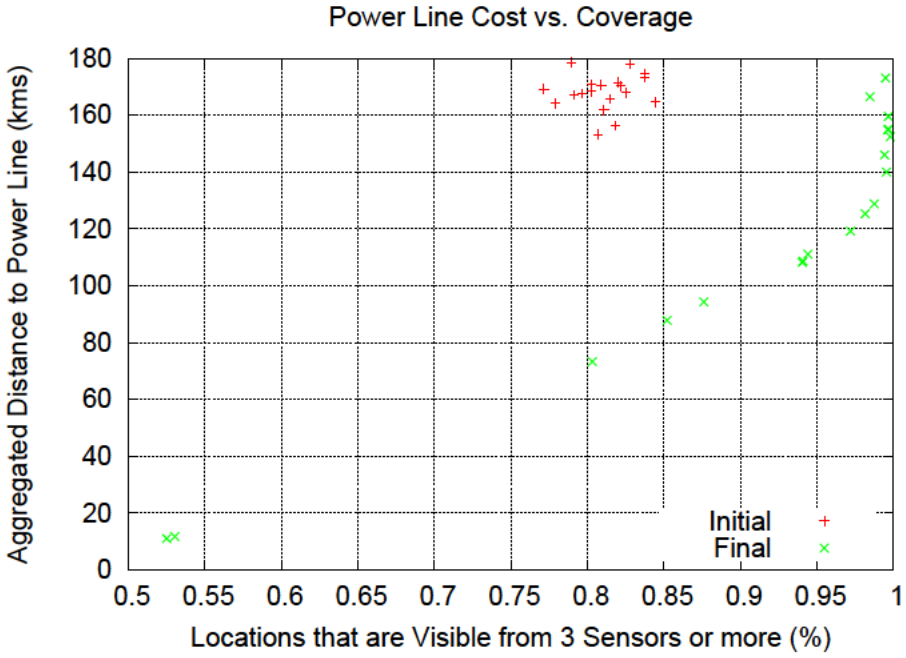


Figure 4.8: Aggregated distance to the power lines (cost) vs. coverage area with $V(x) \geq 3$

The results reveal that the Pareto front approach is useful. The solutions show a compromise between cost and coverage. In the $V(x) \geq 1$ case there is no real advantage over a random seeding in terms of coverage, although the cost drops considerably for 100% coverage. However, in the $V(x) \geq 3$ case, the optimization result is clearly superior to random seeding across the whole Pareto front. At maximum cost, there is a 20% increase in coverage, and at 85% coverage (the best coverage of random seeding) there is a 50% decrease in cost.

Figures 4.9, 4.10 and 4.11 show three sensor deployments in our synthetic scenario in Figure 4.5, for three different choices of $\theta = \{0.1, 0.5, 0.9\}$ (red points represent sensor positions, blue areas zones with less coverage, dark blue areas indicate no coverage at all). The results are consistent with the fact that coverage improves for low θ values. As the values of θ become higher, the sensors tend to concentrate around the power lines. Therefore, power cost decreases, but so the coverage does (Figure 4.11 for $\theta = 0.9$).

Figure 4.12 represents the percentage of grid points covered by different numbers of sensors in the three previous deployments. In the $\theta = 0.9$ case, where the cost function has more weight, over 35% grid points are out of coverage, and, due to the concentration of the sensors around the power lines, there are grid points that are covered by 13 sensors or more. As θ decreases, grid points covered by less than 3 sensors are rare, and the majority of the grid points are covered by 3-5 sensors.

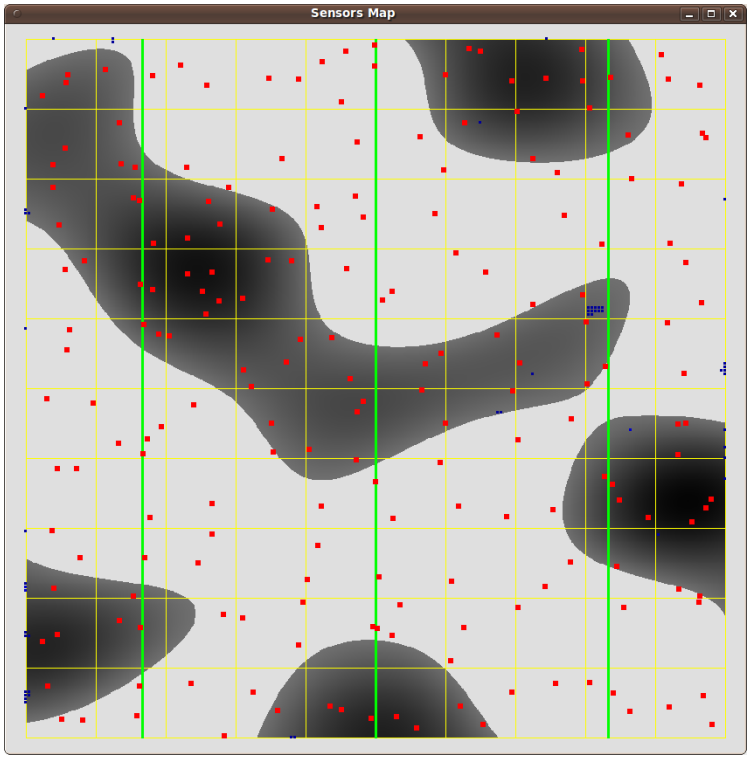


Figure 4.9: A sensor deployment for $\theta = 0.1$

Table 4.2 shows location precision and expected coverage results. The location algorithm described in Section 4.2.2 has been tested in the three previous deployments ($\theta = \{0.1, 0.5, 0.9\}$). Table 4.2 shows the average location error in meters (distance between estimated and real positions) for $10,000$ random gunshot positions in the simulation area, for each θ value and three levels of variation in sensor clocks. These variations are generated by adding a normal Gaussian random toss to the arrival time at each node, for three different values of σ between 0 and 10 ms. The results show that location accuracy increases with the number of detecting nodes, as well as with a better clock synchronization. In addition, this table provides information on the expected coverage at each scenario. For instance, in the $\theta = 0.1$ case, the probability that one or more nodes detect the event is 0.99973, for two or more nodes it is 0.99777, and so on. Clearly, there must be a compromise between detection accuracy, coverage and deployment cost.

4.5.1. Conclusions

In this Chapter, we have proposed a gunshot location procedure based on sensor networks and a Pareto front approach to optimize large scale deployments. The location procedure is based on hyperbolic multilateration using data from a synchronized sensor network. We also propose a practical distributed synchronization

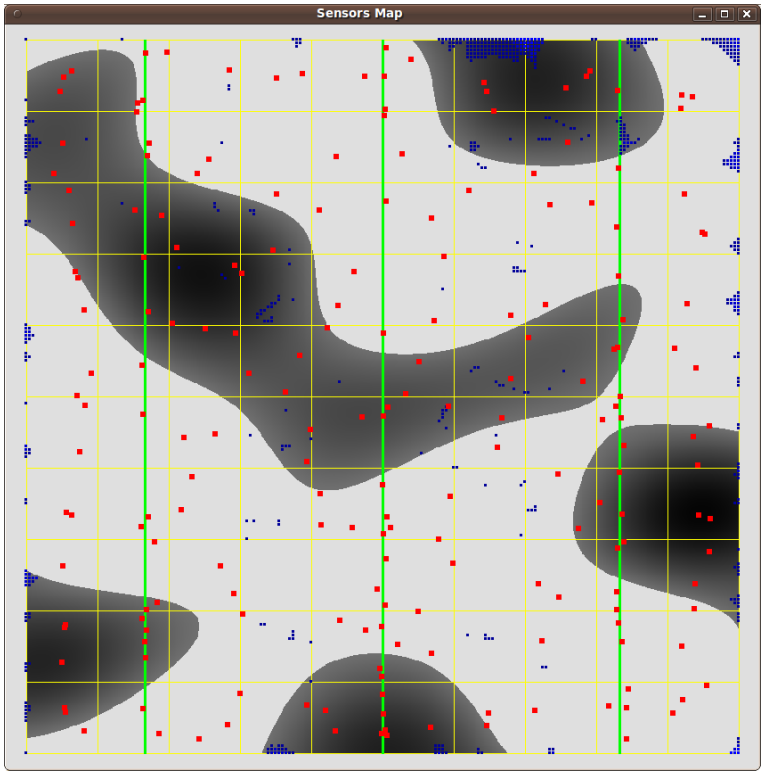


Figure 4.10: A sensor deployment for $\theta = 0.5$

		>1 [0.99973%]	>2 [0.99777%]	>3 [0.99448%]	>4 [0.75258%]	>5 [0.45608%]
$\theta = 0.1$	sync. $< 1ms$	64.40	64.40	64.40	5.30	1.60
	sync. 1ms	78.00	66.90	73.90	5.50	0.00
	sync. 10ms	76.90	90.70	73.40	13.60	3.90
		>1 [0.99988%]	>2 [0.99730%]	>3 [0.98856%]	>4 [0.70991%]	>5 [0.43301%]
$\theta = 0.5$	sync. $< 1ms$	139.20	138.90	123.50	8.30	0.00
	sync. 1ms	146.70	120.20	125.40	22.50	0.00
	sync. 10ms	144.90	162.60	150.20	42.70	4.00
		>1 [0.64476%]	>2 [0.59659%]	>3 [0.57273%]	>4 [0.50632%]	>5 [0.47687%]
$\theta = 0.9$	sync. $< 1ms$	114.00	66.80	20.20	11.00	1.50
	sync. 1ms	99.30	56.40	49.90	19.20	2.40
	sync. 10ms	171.80	127.10	88.00	31.70	19.50

Table 4.2: Gunshot average location error (m)

algorithm for that purpose, with low energy consumption. Sensor network planning follows a Pareto front approach, using a monotone descent method without derivatives that is compatible with realistic optimization functions. Our results are clearly superior to random placement, achieving a 50% cost reduction for 85% coverage. In the two-dimensional scenario, event detection by at least four nodes is required to reach a satisfactory gunshot location accuracy.

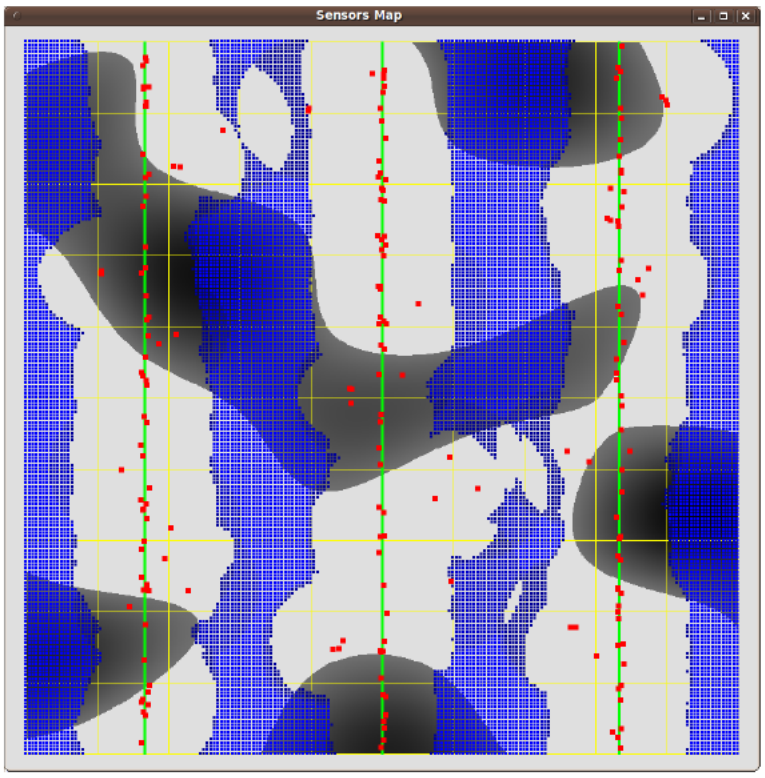


Figure 4.11: A sensor deployment for $\theta = 0.9$

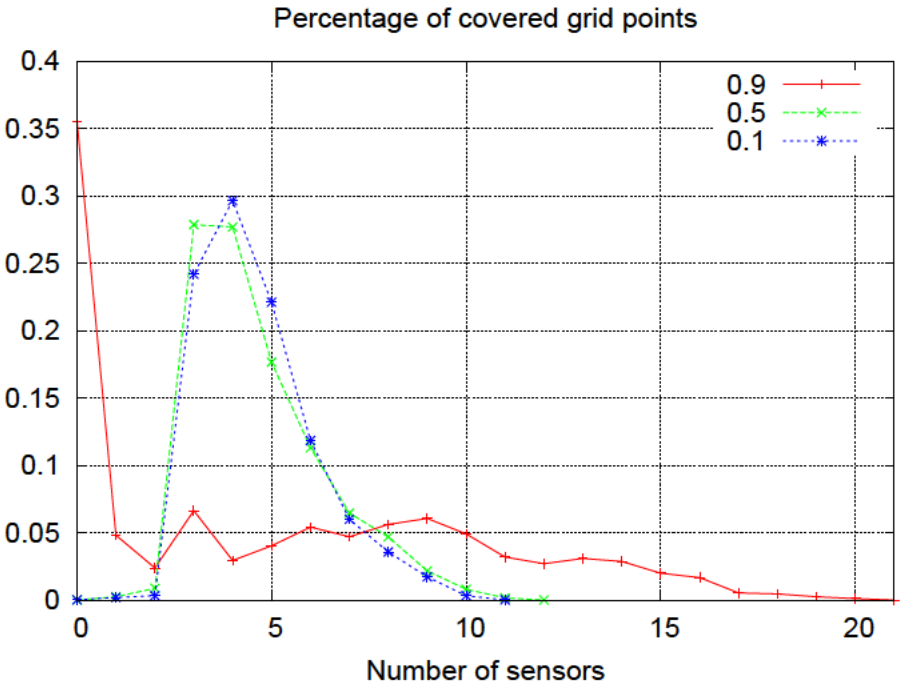


Figure 4.12: Percentage of covered grid points versus number of detecting sensors for different values of θ

Chapter 5

On the optimal random deployment of wireless sensor networks in non-homogeneous scenarios

Riddles are your compulsion. Your addiction. And a riddle that everyone knows the answer to is worthless.

5.1. Introduction

In this Chapter, a special *integrated coverage and connectivity* problem [Ghosh and Das, 2008] is addressed for random deployments. The goal is to find the optimal network deployment scheme that maximizes the joint sensing coverage of the nodes, while satisfying connectivity conditions. Our system model considers the following major features:

1. Node placement is inherently random, due to airborne sensor deployment, for example. Nevertheless, it is assumed that the network designer can select certain placement parameters, which allows some control over the position of the nodes.

2. The nodes are grouped in one or more *clusters*, and each cluster placement can be independently controlled. In this work, Gaussian random distributions are assumed for the clusters, since they correspond to actual observed distributions in airborne seeding [George et al., 2004; Zou and Chakrabarty, 2003; Fang et al., 2006]. For each cluster two parameters can be selected:

the *target point* (i.e. where the nodes are aimed) and the *dispersion* (i.e. the standard deviation). The latter controls how the nodes are spread from such points and may be easily controlled in airborne placements since it is related to the altitude at which the cluster is released (see [George et al., 2004]).

3. Some zones in the target area are more "important" than others (*non-homogeneous* scenario), distinguishing our model from many previous models which assume a uniformly distributed importance across the target area (see Section 5.2). In our model a map describes the importance of each point in the operational region. It is assumed that sensors collect information from a small area (sensing coverage), and that overlapping does not provide additional information. Therefore nodes must be spread to avoid overlapping of sensing areas.

4. Sensed data must be delivered to a control node (*sink* node in WSN terminology), otherwise data will be lost. The routing capabilities of WSN make multi-hop transmissions possible, so sensed data can be delivered whenever a path is available. However, some nodes may be isolated (due to random scattering), and the information they gather cannot reach the sink.

Therefore, the challenge is to select suitable target points and dispersions which provide a trade-off between sensing overlapping and the number of isolated nodes. This leads to a complex stochastic optimization problem. As will be discussed later in Section 5.4, this problem can be efficiently addressed by dividing the optimization into several steps (one per cluster), where simplified placement problems are addressed separately.

Let us remark that, although energy constraints are important in the design of a WSN, this Chapter focuses exclusively on sensing coverage. Those and other constraints will be addressed in the next Chapter.

Without loss of generality, this model is inspired by planetary exploration scenarios such as those proposed in [Gaura and Newman, 2006; Dubowsky et al., 2005], where the operational region is notably larger than the sensing range of a node. So far, interplanetary exploration probes and rovers have only been able to land in and explore small areas in a few solar system bodies such as Mars or Saturn's moon Titan. In this scenario the "important areas" may be, for instance, places where the best chances of finding traces of water exist, as has been the goal of NASA's Mars Sojourn rovers Spirit and Opportunity [Spirit, 2011].

Within this context, WSN may become a feasible alternative for extending monitoring range. Clusters can be deployed over large important areas, and simple nodes would be in charge of data monitoring. Besides, these nodes, rather than deliver the data directly to Earth (which would require complex communication hardware) would relay it via the sink (a special node with such capability). In addition, the Gaussian distribution, selected to model the network placement, also

fits with this scenario. During the terminal descent phase of the probe, the expulsion mechanism can be adjusted to release the nodes at a specific altitude above the surface [Dubowsky et al., 2005].

Major contributions of this work

As far as we know, this is the first work to analytically solve the optimality of a WSN random deployment considering Gaussian clusters. Previous works have mainly focused on network connectivity issues in non-clustered WSNs. Moreover, we also integrate the effect of non-homogeneous importance in the target scenario, unlike previous works which have largely ignored it. Let us remark that our analysis can be also applied to homogeneous scenarios (which is a particular case of the non-homogeneous problem). Section 5.2 discusses these related works.

In summary, the main contributions of this Chapter are:

- We obtain a mathematical optimization problem from the system model (Section 5.3). This problem fully integrates both coverage and connectivity issues for clustered networks.
- We demonstrate how the global problem naturally decomposes into smaller problems, each related to the deployment of a single cluster. These problems can be solved simultaneously, leading to a *global optimization* strategy. As will be discussed later, analytical solutions in this field are still unknown. In this chapter we develop an approximation to characterize the joint sensing coverage of a cluster. Several tests have been performed on this approach and the results (Section 5.5) clearly validate it.
- We show that an approximate *iterative optimization* achieves solutions close to the global optimization solutions in complex scenarios while being computationally feasible for any number of clusters, unlike the global optimization which does not scale up well.
- Finally, we have tested this methodology in complex scenarios, including the lunar Tycho crater as the target scenario (see Figure 5.4) for our validation tests. Providing coverage to such an extension is a challenging project as will be shown in Section 5.5.

5.2. Related work

Deployment optimization focused on either deterministic or random scenarios is a field of increasing interest in WSNs. The problems addressed are related to both communication connectivity and sensing coverage. Optimization of deterministic deployments are surveyed in some works such as [Younis and Akkaya, 2008] and

[Ghosh and Das, 2008]. The rest of this section describes related works in random deployments.

Random placement is a classic problem which was studied long ago for air warfare [Laurent, 1957], yet it is still an active area of research. Many related works such as [E.N., 1961; Ammari and Das, 2008; Isler et al., 2004; Liu and Towsley, 2004] focus on the topology of ad-hoc networks, which are usually modeled by means of *random geometric graphs* [Penrose, 2003] (RGGs). In RGGs, vertices (i.e. nodes) are connected by an edge (i.e. link) if distance between them is below some limit (i.e. communication range). These works consider that in a network it is mandatory to obtain the largest connected component, and they study the conditions under which the *percolation* phenomenon occurs. This refers to the critical point in the average number of connections per node for which most nodes belong to same component (giant component), leading to a fully connected network.

[E.N., 1961] first studied this issue to find the critical density of a Poisson Process at which a network can provide long-distance multi-hop communications. [Ammari and Das, 2008] analyze the degree of sensing coverage and connectivity of a WSN, assuming that sensors are deployed following a Poisson point distribution. The authors discuss how node density affects network connectivity. The main finding of [Ammari and Das, 2008] was the establishment of a relationship between coverage and connectivity for this kind of network deployment. Percolation and random geometric graph theory are also utilized in [Isler et al., 2004] to investigate the density of nodes required to guarantee coverage and connectivity in an uniform deployment of sensors. The authors present two incremental deployment algorithms which try to improve the placement of sensors in a series of consecutive deployments. [Liu and Towsley, 2004] characterize, by means of simulations, the fundamental coverage properties of a WSN. All these works aim at selecting basic network properties such as node density, communication range, and so forth, and their results are applicable to large scale *non-clustered* networks. In our work we also investigate connectivity and coverage, but we focus on clustered Gaussian deployments, rather than on uniform or Poisson point processes. As stated in the introduction, Gaussian clusters are the realistic models for airborne deployments. As will be shown in the next sections, unlike previous works, our aim is not to find conditions for percolation (full–connectivity of the network) but to maximize the sensing coverage of the cluster (which may occur in a sub-critical connection regime).

Some other works have also considered random Gaussian positions for the nodes. For example, [Leoncini et al., 2009] consider a gridded area where a cell is considered to be covered if a sensor falls inside this cell. Like our work, sensor position randomness is modeled with a Gaussian distribution. The authors develop a deployment strategy such that a certain degree of coverage is met and the total number of deployed nodes is minimized. The results suggest that the best alterna-

tive for deployment is to aim the sensors at the center of each cell. However, this work does not address the general connectivity problem in the network, and uses a simplified coverage model. [Zou and Chakrabarty, 2003] present two optimization algorithms which try to cover the vertices of a gridded surface with the minimum number of sensors. Similarly to our work, they consider a Gaussian model to describe non-deterministic placement but they do not consider the connectivity of the network. [Wang et al., 2008] introduce two algorithms for Gaussian deployments which reduce the number of sensors while satisfying the required coverage and lifetime conditions. The authors use results from uniform random deployments to develop the connectivity model. As a drawback, their model can be applied only in scenarios of reduced dimensions with a high density of nodes. [Ishizuka and Aida, 2004] investigate three functions to distribute sensor nodes over a surface to determine their respective fault tolerance properties. The comparison is performed with Gaussian, Uniform, and R-random distributions. In R-random distributions nodes are scattered uniformly with respect to radial and angular directions. Simulation results show that Gaussian and R-random placement strategies outperform the uniform patterned strategy because they tend to concentrate more nodes close to the sink. Besides, some recent studies [Bari et al., 2008; Xu et al., 2010; Bari et al., 2011] have analyzed topologies formed by two-tier hierarchical clusters, which improve network lifetime and cover large surfaces easily. All these works develop and analyze the performance of different placement heuristics. By contrast, in our approach we develop an *optimal* stochastic program from the network model which allows us to directly compute the best cluster configurations. In Section 5.5 we show how our solution outperforms a reference heuristic.

In addition, different works have also proposed optimization problems for placement problems, but they are more limited than our approach and do not focus on either random Gaussian clusters or on joint sensing coverage maximization. For example, [Krause et al., 2008] propose a combinatorial optimization problem to maximize mutual information. [Wang et al., 2006] present an optimization model to identify the sensor locations that maximize the average probability of event detection per point. In this case, the search space is limited to the positions represented by a grid. This work is similar to ours because it attempts to retrieve maximum importance, but in our study we do not restrict the positions of candidate sensors.

Finally, most of the previous approaches do not analyze the impact of scenarios with non-homogeneous importance. [Huang and Tseng, 2005] mention this issue and propose an algorithm for covering important areas with more sensors. [Poduri and Sukhatme, 2004] provide a self-deployment strategy for robots which autonomously move to better positions. [Dhillon et al., 2002] discuss the inherent uncertainty of sensor readings and assign a probability measure to sensor detections. These probabilities help to model terrain effects. The authors also describe an algorithm for placing the sensors into a grid in order to provide adequate coverage. [Aitsaadi

et al., 2010] introduce a multi-objective deployment algorithm (MODA) which takes into consideration the following constraints: deployment cost (number of sensors), event detection probability, connectivity, and energy consumption. The proposed algorithm minimizes energy consumption and the number of sensors required to accomplish coverage goal. Similarly to this Chapter, the deployment scenario has non-homogeneous sensing requirements. [Dasgupta et al., 2003] propose a scenario in which a certain amount of data has to be gathered by the network. The data is non-uniformly distributed and the authors propose a heuristic that relocates nodes in order to balance the load of data transmitted from zones with greater density of data, and removes sensors in zones with less density of data. [Isler et al., 2004] modeled coverage with a weighting function, which assigns a reward to each point covered that is inversely proportional to the distance from the sensor. Sensor positions that are likely to increase the total weight are preferred. None of these works address the placement problem from the perspective of optimizing clustered Gaussian networks.

Summarizing, previous works in random deployments have largely ignored both the clustered nature of WSNs and the non-homogeneous distribution of *importance*. Ours is the first to develop an integrated model comprising coverage and network connectivity of Gaussian clusters. The next sections describe our optimization problem from a theoretical perspective and provides analytical tools to solve it.

5.3. Sensor Positioning Model

In this section we characterize the deployment scenario and the model of our network, and describe the underlying assumptions which define the constrained optimization problem.

Our model comprises M independent *clusters* of nodes, each composed by N sensors (including a sink node). These clusters must be deployed in a non-homogeneous target area, such that the importance (quality) of the information gathered is maximized. The planning is subject to communication constraints, since nodes must have a path (either direct or via a multi-hop route) to their corresponding sink node.

Figure 5.1 illustrates the main concepts of the model. The deployment area is a euclidean space $\mathcal{X} \subseteq \mathbb{R}^2$ (with distance function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$). Besides, there exists a value of importance associated with every $x \in \mathcal{X}$ which is defined by a real and continuous *importance function* $\alpha : \mathcal{X} \to \mathbb{R}_0^+$. Therefore, the total importance contained in the target area can be computed as:

$$\Gamma_{max} = \int_{\mathcal{X}} \alpha(x) \mathrm{d}x \tag{5.1}$$

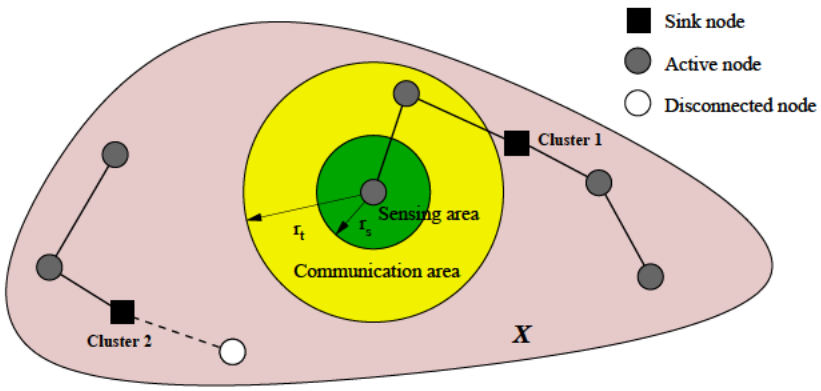


Figure 5.1: System model

Let $\Phi_j = \{\varphi_i^j : i = 1, \ldots, N\}$ denote the nodes in cluster j for $j = 1, \ldots, M$, and $\mathcal{S}_j = \{x_i^j : i = 1, \ldots, N\}$ are the sets which contain the positions of the nodes in the target area.

For the sensor network architecture the following considerations have been established:

- The sensing hardware is homogeneous, i.e. sensor nodes are of the same type and have the same sensing capabilities.
- Each cluster operates independently, i.e. nodes of different clusters do not communicate.
- The network sink can be any of the N sensor nodes. Without loss of generality, we assume that nodes φ_1^j for $j = 1, \ldots, M$ are the sinks and x_1^j for $j = 1, \ldots, M$ their positions.
- The *transmission range* r_t is the longest distance between two mutually communicating nodes.
- The *sensing range* r_s is the longest distance between the node and the locations which it is able to sense. It is assumed that a sensor is able to capture all the importance within its sensing range. Therefore the importance covered by sensor φ_i^j is:

$$\int_{B(x_i^j, r_s)} \alpha(x) \mathrm{d}x \tag{5.2}$$

  being $B(x, r)$ the open ball in $\mathbb{R}^2$ centered in x with radius r.
- The set $\Phi_j^\star \subseteq \Phi_j$ for $j = 1, \ldots, M$, contains the connected cluster of nodes able to transmit sensed data to their corresponding sink φ_1^j, either directly or

by multi-hop transmissions. Set $\Phi_j^\star$ includes the sink itself. Hereafter, let us refer to the nodes belonging to $\Phi_j^\star$ as *active* nodes.

In our model it is assumed that, for each cluster j, nodes are spread randomly around the target point of this cluster (thereby denoted as x_*^j), following independent Gaussian distributions $N(0, \sigma_j)$ in each axis. Let us denote Δ_j as the bivariate Gaussian distribution which measures this displacement. Then, the position of the nodes in the cluster j is given by independent identically distributed random variables $x_i^j = x_*^j + \Delta_j$ for $i = 1, \ldots, N$.

We selected Gaussian distributions for some of their particular properties. First, it has been demonstrated (see [Jaynes and Bretthorst, 2003; Gowers et al., 2008]) that in the experiment of throwing an object aimed at a particular location, the error can be modeled as independent Gaussian variables in each axis. The resulting density function depends on the distance from the target point, but not on its coordinates (rotationally invariant property). This property can be applied to predict the position of every sensor node in a cluster, aimed at the same target point, like in our model. Moreover, Gaussian dispersion can be modulated in an airborne deployment, since the final position error is correlated with the cluster releasing altitude (see [George et al., 2004]). Thus, the expulsion mechanism can be adjusted to release the nodes at an specific altitude above the surface, providing certain control over the final position of the nodes.

Finally, let us assume that the importance captured from an open region $\mathcal{A} \subseteq \mathcal{X}$ is maximal if for all $a \in \mathcal{A}$ there exists at least one active node $\varphi_i^j \in \Phi_j^\star$ for some $j \in [1, \ldots, M]$ such that $a \in B(x_i^j, r_s)$. In other words, if the sensing areas of active nodes overlap, the importance captured in those areas does not increase. Therefore, the open set describing the sensed area is:

$$\mathcal{B} = \left[\bigcup_{j=1}^{M} \bigcup_{i \in \Phi_j^\star} B(x_i^j, r_s) \right] \bigcap \mathcal{X} \tag{5.3}$$

Let us remark that since node position is random, $\mathcal{B}$ is random as well. Therefore the importance captured in a network deployment is also random. The next section proposes a stochastic mathematical program for our scenario.

5.4. Optimization model

The question addressed in this section is how to spread the sensors over the designated areas to maximize the expected covered importance.

Let $1_{\mathcal{B}}(x)$ denote the indicator function equal to 1 if $x \in \mathcal{B}$ and 0 otherwise. The optimization problem describing our model can be stated as:

Given

$$\alpha : \mathcal{X} \rightarrow \mathbb{R}_0^+$$

find

$$x_*^j, \sigma_j, \text{ for } j = 1, \ldots, M$$

such that

$$\max_{\substack{x_*^j, \sigma_j \\ j=1,\ldots,M}} \mathbb{E}\{\int_{\mathcal{X}} \alpha(x) 1_{\mathcal{B}}(x) \mathrm{d}x\} \tag{5.4}$$

subject to

$$x_*^j \in \mathcal{X}$$
$$\sigma_j > 0$$

for $j = 1, \ldots, M$.

Let us consider an example area with non-homogeneous importance, like the one selected for the sample scenario shown in Figure 5.4. Clearly, in large/complex scenarios manual placement is not possible. Besides, direct algorithmic computation of the previous optimization problem is unfeasible: note that Eq. (5.4) requires the calculation of $\mathbb{E}\{1_{\mathcal{B}}(x)\} = Pr[x \in \mathcal{B}]$ given any particular combination of network configuration parameters x_*^j, σ_j, for j=1, ..., M. If all the nodes in a cluster belong to its active set, this probability can be easily derived (following the method used later in Section 5.4.1). However, this is an oversimplification since $\mathcal{B}$ is the area covered *only by active nodes*. The actual probability computation leads to a connectivity problem which falls in the field of percolation theory in RGGs, but general results for Gaussian clusters are still unknown [Penrose, 2003]. Therefore, in this Chapter, we propose an approximated computation of Eq. (5.4) using a mixed analytical-numerical approach which is described in the following parts of the Chapter.

In order to find a suitable optimization model let us recall that clusters operate independently. This suggests that the whole expected importance can be computed as the sum of the expected importances captured by each separate cluster. In fact, formula (5.4) can be rewritten as:

$$\max_{\substack{x_*^j, \sigma_j \\ j=1,\ldots,M}} \mathbb{E}\{\int_{\mathcal{X}} \alpha(x) 1_{\bigcup_{j=1}^{M} \mathcal{B}^j}(x) \mathrm{d}x\} \tag{5.5}$$

where $\mathcal{B}^j$ is the area covered by the cluster j, that is,

$$\mathcal{B}^j = \bigcup_{\varphi_i^j \in \Phi_j^\star} B(x_i^j, r_s)$$

Let $0_{\mathcal{B}}(x)$ denote the inverse indicator function equal to 0 if $x \in \mathcal{B}$ and 1 otherwise. Reorganizing formula (5.5):

$$\begin{aligned} \max_{\substack{x_*^j, \sigma_j \\ j=1,\dots,M}} \mathbb{E}\{ & \int_{\mathcal{X}} \alpha(x) 1_{\mathcal{B}^1}(x) \mathrm{d}x + \int_{\mathcal{X}} \alpha(x) 0_{\mathcal{B}^1}(x) 1_{\mathcal{B}^2}(x) \mathrm{d}x + \dots \\ & \dots + \int_{\mathcal{X}} \alpha(x) \prod_{j=1}^{M-1} 0_{\mathcal{B}^j}(x) 1_{\mathcal{B}^M}(x) \mathrm{d}x \} \end{aligned} \tag{5.6}$$

In words, for iteration j the area already covered by previous clusters must be subtracted (multiplying $\alpha(x)$ by $\prod_{j=1}^{j-1} 0_{\mathcal{B}^j}(x)$). The previous equation can be written more compactly by denoting:

$$\begin{aligned} \alpha^j(x) &= \alpha^{j-1}(x) 0_{\mathcal{B}^{j-1}}(x), \text{ if } j > 1 \\ \alpha^1(x) &= \alpha(x) \end{aligned}$$

Thus, $\alpha^j(x)$ represents the *remaining* importance at iteration j. Using this notation and since the expectation is a linear operator, the problem is to:

$$\max_{\substack{x_*^j, \sigma_j \\ j=1,\dots,M}} \sum_{j=1}^{M} \mathbb{E}\{ \int_{\mathcal{X}} \alpha^j(x) 1_{\mathcal{B}^j}(x) \mathrm{d}x \} \tag{5.7}$$

By applying Fubini's theorem the order of integration and expectation (note that in a continuous domain this is actually a finite integral) can be changed, and leads to:

$$\max_{\substack{x_*^j, \sigma_j \\ j=1,\dots,M}} \sum_{j=1}^{M} \int_{\mathcal{X}} \mathbb{E}\{ \alpha^j(x) 1_{\mathcal{B}^j}(x) \} \mathrm{d}x \tag{5.8}$$

The expectation inside the integral of the last formula can be decomposed as:

$$\mathbb{E}\{ \alpha^j(x) 1_{\mathcal{B}^j}(x) \} = \mathbb{E}\{ \alpha(x) \prod_{k=1}^{j-1} 0_{\mathcal{B}^k}(x) 1_{\mathcal{B}^j}(x) \} \tag{5.9}$$

Note that for any $j=1,\dots,M$, the random sets $\mathcal{B}^k$, for $k=1,\dots,j$ are independently determined. Thus, the random variables $0_{\mathcal{B}^k}(x)$, for $k=1,\dots,j-1$ are also

mutually independent and independent of $1_{\mathcal{B}^j}(x)$. Hence,

$$\mathbb{E}\{\alpha^j(x)1_{\mathcal{B}^j}(x)\} = \alpha(x)\prod_{k=1}^{j-1}\mathbb{E}\{1_{\mathcal{B}^j}(x)\mathbb{E}\{0_{\mathcal{B}^k}(x)\}\} \tag{5.10}$$

Besides,

$$\mathbb{E}\{1_{\mathcal{B}^j}(x)\} = Pr[x \in \mathcal{B}^j] = 1 - \mathbb{E}\{0_{\mathcal{B}^j}(x)\}$$

Finally, the problem is expressed as:

$$\max_{\substack{x_*^j,\sigma_j \\ j=1,\ldots,M}} \sum_{j=1}^{M}\int_{\mathcal{X}}\mathbb{E}\{\alpha^j(x)\}Pr[x \in \mathcal{B}^j]\mathrm{d}x \tag{5.11}$$

where,

$$\mathbb{E}\{\alpha^j(x)\} = \alpha(x)\prod_{k=1}^{j-1}Pr[x \notin \mathcal{B}^k]$$

Thus, we have proved that the problem is reduced to smaller problems each related to a single cluster. The solution to these individual problems is analyzed in the next section.

5.4.1. Single cluster analysis

This section addresses the problem of determining the probability that a point $x \in \mathcal{X}$ is covered by active nodes of a given cluster j, *i.e.* computing $Pr[x \in \mathcal{B}^j]$. To simplify the notation, the script j referring to cluster j is omitted through this section. The strategy to obtain this probability is to approximate the importance captured by a cluster (let Γ denote such importance) and to then relate it to the analytical importance:

$$\int_{\mathcal{X}}\alpha(x)Pr[x \in \mathcal{B}]\mathrm{d}x \tag{5.12}$$

To start the computation of Γ let us recall from Section 5.3 that the position of each sensor in a cluster is an independent identically distributed bivariate random variable $x_* + \Delta$ whose joint probability density function is given by:

$$f_{x_*+\Delta}(x; x_*, \sigma) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{d(x, x_*)^2}{2\sigma^2}\right).$$

A node placed at position x with r_s sensing range will then capture the importance:

$$\int_{\mathcal{B}(x,r_s)} \alpha(x)\mathrm{d}x \approx \pi r_s^2 \alpha(x) \tag{5.13}$$

Approximation holds if importance is uniform for scales comparable to r_s. Since the sensing ranges are usually small and the terrain does not change abruptly (note that $\alpha(x)$ is assumed to be continuous) approximation in Eq. (5.13) is taken in this work. Therefore, the *expected* importance I gathered by a single sensor is:

$$I \approx \int_{\mathcal{X}} \pi r_s^2 \alpha(x) f_{x_*+\Delta}(x; x_*, \sigma)\mathrm{d}x \tag{5.14}$$

At first glance the expected importance captured by the N nodes in the cluster could be estimated as $N \times I$. This approach is equivalent to assuming that all the information can be delivered to the sink and that the sensing areas of all the nodes are non-overlapping. However, this is an oversimplification as discussed before. Figure 5.2 shows some random deployments aimed at position (0,0) where σ has values 5, 15, 25, and 35, $r_s = 3$ and $r_t = 10$. Active nodes are depicted with their sensing areas (circles) and disconnected nodes are marked with '+' symbols. Clearly, for low σ values most nodes are connected but sensing areas overlap, and thus deployment is inefficient. On the other hand, for high values of σ the deployment is also inefficient because of the sub-connected topology. In our example, the best configurations are achieved for mid dispersions.

Therefore, the goal is to select a balanced σ that keeps connectivity as high as possible while avoiding overlapping of sensing areas. To characterize this problem, let Ω denote the average number of active nodes in the cluster. In addition, let θ denote the ratio of overlapped sensing areas. Then, the captured importance of the cluster, Γ, can be computed as:

$$\Gamma \approx I(\Omega(1-\theta)+\theta) \tag{5.15}$$

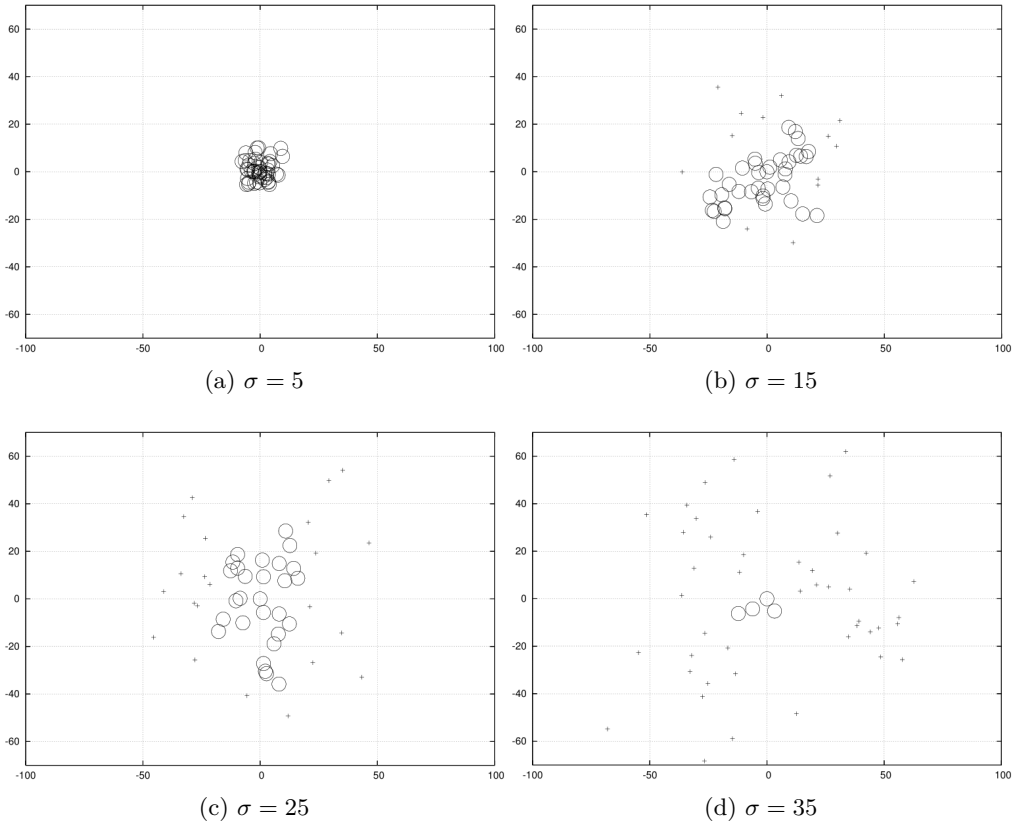
Note that this is an approximation. If $\theta = 0$ (no overlapping), then the Ω nodes capture on average $I \times \Omega$. On the other hand, if the covered areas of all the nodes overlap, then $\theta = 1$, *i.e.* the captured importance corresponds to a single node (I). Moreover, note that Ω and θ depend on σ but are independent of the cluster's central position x_*. However, I depends on both σ and x_*.

Therefore, we obtain:

$$I(\Omega(1-\theta)+\theta) \approx \int_{\mathcal{X}} \alpha(x) Pr[x \in \mathcal{B}]\mathrm{d}x \tag{5.16}$$

Substituting I by the approximation of Eq. (5.14):

$$\int_{\mathcal{X}} \pi r_s^2 \alpha(x) f_{x_*+\Delta}(x; x_*, \sigma)(\Omega(1-\theta)+\theta)\mathrm{d}x \approx \int_{\mathcal{X}} \alpha(x) Pr[x \in \mathcal{B}]\mathrm{d}x \tag{5.17}$$



(a) $\sigma = 5$ (b) $\sigma = 15$

(c) $\sigma = 25$ (d) $\sigma = 35$

Figure 5.2: Effect of sensor dispersion in the importance covered. Importance is captured only in the circled areas

Hence, the probability $Pr[x \in \mathcal{B}]$ is:

$$Pr[x \in \mathcal{B}] \approx \pi r_s^2 f_{x_*+\Delta}(x; x_*, \sigma)(\Omega(1-\theta)+\theta) \tag{5.18}$$

The next sections discuss the computation of Ω and θ, respectively.

5.4.1.1. Ω computation

Reference [Penrose, 2003] contains analytical descriptions related to the connectivity regime for RGGs with vertices distributed uniformly. Disappointingly, similar results for normally distributed RGGs are still unknown, and the analytical computation of Ω is not possible. To overcome this issue, Ω has been numerically evaluated in this work as a function of N and $\gamma{=}p_{r_t}\sqrt{N}$. This last parameter is related to the thermodynamic limit in RGGs, and it is commonly used to study connectivity regimes in random graphs. Besides, p_{r_t} is the probability that two nodes are connected, or, in other words, that the distance between them is less than r_t. It can be easily calculated (see [Vales-Alonso et al., 2007]):

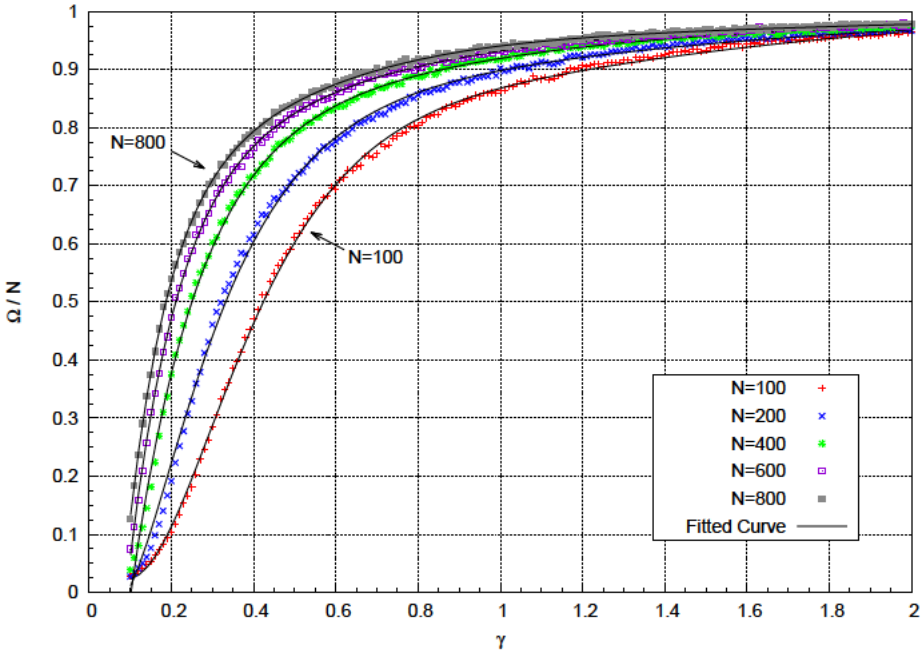


Figure 5.3: $\frac{\Omega}{N}$ curves

$$p_{r_t} = Pr[d(x_i, x_j)^2 \leq r_t^2] = 1 - \exp\left(-\frac{r_t^2}{4\sigma^2}\right) \tag{5.19}$$

Therefore,

$$\gamma = \frac{1 - \exp\left(-\frac{r_t^2}{4\sigma^2}\right)}{\frac{1}{\sqrt{N}}} \tag{5.20}$$

For each configuration of N and γ, 10000 instances of the random network have been selected. For each instance, the order of the component containing the sink has been computed (using Dijkstra's algorithm to determine which nodes have connectivity) obtaining $\frac{\Omega}{N}$, that is, the ratio of connected nodes in the cluster. Finally, the samples from each instance are averaged. We took values ranging from N=100 to N=800, and from γ=0.1 to γ=$\frac{1}{\sqrt{N}}$. The results are shown in Figure 3. Indeed, since in the optimization model the variable σ (and therefore γ) is continuous, it is desirable to have curves interpolating our numerical evaluations. We then fitted the curves using an exponential model:

$$\frac{\Omega}{N}(\gamma) = 1 + a\, e^{-b\,\gamma} + c\,\gamma\, e^{-d\,\gamma} + f\,\gamma^2\, e^{-g\,\gamma} \tag{5.21}$$

The coefficients which provides the minimal square error fit are shown in Table 5.1.

N	a	b	c	d	f	g
100	-1.014	16.54	-14.61	6.367	-1.327	2.517
200	-0.8995	3.189	-7.634	8.334	-0.5282	2.131
400	-1.709	5.891	-0.7896	2.485	-0.02944	1.067
600	-1.846	7.907	-1.455	3.471	-0.09645	1.476
800	-1.898	8.371	-0.1461	1.3	-6.898	5.891

Table 5.1: $\frac{\Omega}{N}$ fitting coefficients of $1 + a\, e^{-b\,\gamma} + c\,\gamma\, e^{-d\,\gamma} + e\,\gamma^2\, e^{-f\,\gamma}$

5.4.1.2. θ computation

Recall that θ represents the ratio of overlapped sensing area in a cluster of N nodes distributed normally around a target point x_*. Clearly, θ does not depend on the central point x_*. Therefore, and without loss of generality, $x_* = (0,0)$ throughout this section and, thus, $x_i = \Delta$ for all $i = 1, \ldots, N$. Let A denote the area sensed by *all* the nodes in the cluster (which is a random value). Note that the maximum area which can be covered is $N\pi r_s^2$. Therefore,

$$\theta = 1 - \frac{\overline{A}}{N\pi r_s^2} \tag{5.22}$$

In order to compute θ, let $p(x)$ denote the probability that a point $x \in \mathcal{X}$ belongs to the area sensed. Then,

$$\begin{aligned} p(x) &= 1 - Pr[x \notin \cup_{i=1}^{N} B(x_i, r_s)] \\ &= 1 - \prod_{i=1}^{N} (1 - Pr[x \in B(x_i, r_s)]) \\ &= 1 - (1 - Pr[x \in B(\Delta, r_s)])^N \end{aligned} \tag{5.23}$$

Moreover,

$$Pr[x \in B(\Delta, r_s))] = \int_{x' \in B(x, r_s)} \frac{1}{2\pi\sigma^2} e^{-\frac{\|x'\|^2}{2\sigma^2}} \mathrm{d}x'$$

For a small $\frac{r_s}{\sigma}$ ratio this equation can be approximated as:

$$Pr[x \in B(\Delta, r_s))] \approx \pi r_s^2 f_\Delta(x) \tag{5.24}$$

Where $f_\Delta(x)$ represents the density of probability of the Gaussian random variable Δ in x. Thus, and using the binomial theorem:

$$\begin{aligned} p(x) &\approx 1-(1-\pi r_s^2 f_\Delta(x))^N \\ &= 1-\sum_{i=0}^{N}\binom{N}{i}(-1)^i(\pi r_s^2 f_\Delta(x))^i \\ &= \sum_{i=1}^{N}\binom{N}{i}(-1)^{i+1}(\pi r_s^2 f_\Delta(x))^i \end{aligned} \tag{5.25}$$

Then, the expected covered surface ($\overline{A}$) of the cluster of N nodes is computed by integrating $p(x)$ over $\mathbb{R}^2$:

$$\begin{aligned} \overline{A} &= \int_{x\in\mathbb{R}^2}\sum_{i=1}^{N}\binom{N}{i}(-1)^{i+1}(\pi r_s^2 f_\Delta(x))^i \mathrm{d}x \\ &= \sum_{i=1}^{N}\binom{N}{i}(-1)^{i+1}\int_{x\in\mathbb{R}^2}\frac{(\pi r_s^2)^i}{(2\pi\sigma^2)^i}e^{-\frac{i(\|x\|^2)}{2\sigma^2}}\mathrm{d}x \\ &= \sum_{i=1}^{N}\binom{N}{i}(-1)^{i+1}\frac{(r_s^2)^i}{(2\sigma^2)^{i-1}}\frac{\pi}{i} \end{aligned} \tag{5.26}$$

Hence, θ is given by:

$$\theta = 1-\frac{\overline{A}}{N\pi r_s^2} = 1-\sum_{i=1}^{N}\binom{N}{i}(-1)^{i+1}\left(\frac{r_s^2}{2\sigma^2}\right)^{i-1}\frac{1}{i}\frac{1}{N} \tag{5.27}$$

5.4.2. Statement of the global optimization

Therefore, from Eqs. (5.11), (5.18), and (5.4) the global problem can be stated as solving:

$$\max_{\substack{x_*^j,\sigma_j \\ j=1,\ldots,M}} \sum_{j=1}^{M}\int_{\mathcal{X}}\mathbb{E}\{\alpha^j(x)\}\pi r_s^2 f_{x_*^j+\Delta_j}(x;x_*^j,\sigma_j)(\Omega_j(1-\theta_j)+\theta_j)\mathrm{d}x \tag{5.28}$$

where

$$\mathbb{E}\{\alpha^j(x)\} = \alpha(x)\prod_{k=1}^{j-1}(1-\pi r_s^2 f_{x_*^k+\Delta_k}(x;x_*^k,\sigma_k)(\Omega_k(1-\theta_k)+\theta_k))$$

where Ω and θ are given respectively by Eqs. (5.21) and (5.27). And subject to

$$x_*^j \in \mathcal{X}$$
$$\sigma_j > 0$$

for $j = 1, \ldots, M$.

This global version of the problem involves the *simultaneous* computation of all the variables. As will be shown in Section 5.5 this approach does not scale up well in memory. An alternative approach is to solve each cluster deployment separately. That is, after selecting a solution for a cluster this solution does not change. This leads to the iterative optimization version, described in the following section.

5.4.3. Statement of the iterative optimization

In order to avoid scalability issues the problem can be divided into a sequence of steps; at each step, the optimization of just a single cluster if considered. That is, at step j only the cluster j configuration x_*^j, σ_j is selected. Once fixed, the configuration will not vary for the next steps. At each step the map needs to be corrected to discount the effect of previous clusters. This can be done easily, since $\mathbb{E}\{\alpha^j(x)\})$ is the expected map with the remaining importance.

Thus, the following iterative optimization procedure is used:

1. Set $\mathbb{E}\{\alpha^1(x)\}$=$\alpha(x)$
2. For j=$1, \ldots, M$
   a) Given the importance map $\mathbb{E}\{\alpha^1(x)\}$, solve the optimization problem:

$$\max_{x_*^j, \sigma_j} \int_{\mathcal{X}} \mathbb{E}\{\alpha^j(x)\} \pi r_s^2 f_{x_*^j + \Delta_j}(x; x_*^j, \sigma_j)(\Omega_j(1-\theta_j) + \theta_j) \mathrm{d}x \qquad (5.29)$$

   subject to

$$x_*^j \in \mathcal{X}$$
$$\sigma_j > 0$$

   b) Compute $\mathbb{E}\{\alpha^{j+1}(x)\}$=$\mathbb{E}\{\alpha^j(x)\}[1-\pi r_s^2 f_{x_*^j + \Delta_j}(x; x_*^j, \sigma_j)(\Omega_j(1-\theta_j)+\theta_j)]$ as the updated importance map for step $j+1$.

This leads to:

$$\sum_{j=1}^{M} \max_{x_*^j, \sigma_j} \int_{\mathcal{X}} \alpha^j(x) \pi r_s^2 f_{x_*^j + \Delta_j}(x; x_*^j, \sigma_j)(\Omega_j(1-\theta_j) + \theta_j) \mathrm{d}x \qquad (5.30)$$

Note that here the sum sign is outside the optimization operator, unlike the situation in formula (5.28). Therefore this approach may not achieve globally optimal solutions. Instead, iterative optimization only finds optimal cluster configurations for a given step j. The rationale is that having a cluster capture high importance at a given step may lead to "bad" clusters capturing low importance in subsequent steps. This is particularly true if the area to cover is small compared to the size of the cluster. Section 5.5 provides examples of this kind.

However, in scenarios in which the area is much larger than the size of the cluster, this effect is less noticeable since the best solution tends to separate clusters because areas already covered possess less importance. This effect is also shown in the examples provided in Section 5.5. Besides, the iterative optimization version has the advantage of being scalable to any number of clusters.

5.5. Deployment examples and results

The main scenario selected to test the performance of our optimization methodology is the lunar crater Tycho (Figure 5.4). Tycho is a prominent lunar impact crater located in the southern lunar highlands. Figure 5.4b illustrates an image of Tycho (35×100 Km^2) that the Clementine mission [Clementine, 2011] mapped onto the Moon's surface. The resulting size of the image was 392×1100 pixels. This color scale mosaic allows different rock units to be distinguished and shows the location of fresh, "mafic" material (i.e., materials relatively rich in iron and magnesium). The importance selected ranges from 0 (minimum) to 255 (maximum), and is related to the presence of these materials.

Let us remark that the optimization procedure is *independent* of the importance map selected and of the network parameters N, r_t and r_s. Therefore, no changes are required to compute the best placement for a different configuration. Our optimization algorithm has been implemented using the *General Algebraic Modeling System* (GAMS) [Brooke et al., 1998], a high-level modeling system for mathematical programming and optimization, and CONOPT [Drud, 1985], a solver for large-scale nonlinear optimization. GAMS allowed us to define our optimization problem directly from the mathematical descriptions provided in Sections 5.4.2 and 5.4.3.

Let us recall that in the global optimization case the GAMS algorithm is executed to simultaneously find the optimal target point (x_*) and dispersion (σ) for the M clusters (see Section 5.4.2), whereas for the iterative case a GAMS instance is consecutively run M times using modified importance maps (see Section 5.4.3). In both cases, the CONOPT solver proceeds iteratively until the algorithm eventually converges to an optimum. Besides, the solver requires initial values for the variables x_* and σ (M initial pairs in the global case) to start computations, and the final solution is not guaranteed to be a global optimum solution, but rather a local

		Γ(%)	
		Analytical	Experimental
N=800	$r_s = 4,\ r_t = 12$	53.2%	49.1%
N=600	$r_s = 4,\ r_t = 12$	44.4%	41.4%
N=400	$r_s = 4,\ r_t = 12$	33.8%	31.5%
N=200	$r_s = 4,\ r_t = 12$	20.5%	19.4%
N=100	$r_s = 4,\ r_t = 12$	12.3%	11.8%

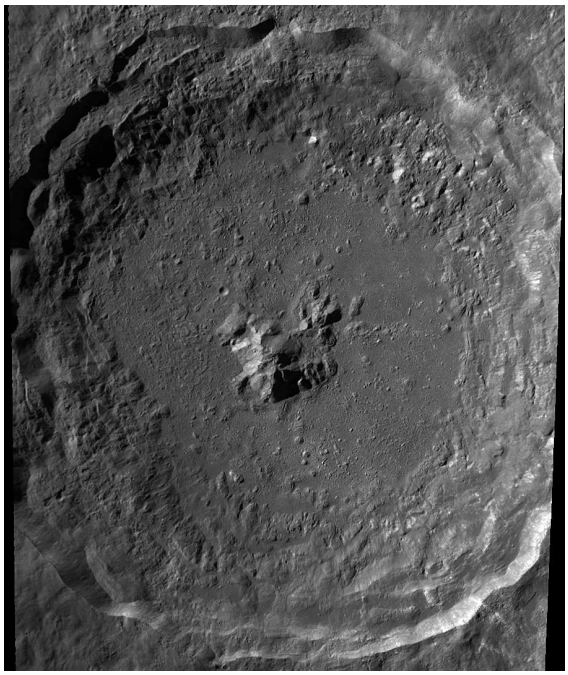
Table 5.2: Analytical vs experimental values computed averaging 5000 experiments, Tycho map for M=10

one. We observed large variations in the solution depending on the initial positions x_*. To mitigate this we ran each optimization problem with several initial values and selected the best one. In all the examples in this section the global method considers 75 randomly chosen initial cluster placements. In the iterative method we fixed 36 different initial x_* per step, as shown in Figure 5.4c. Let us remark that the overall number of runs in the iterative case is then $M\times$36, and therefore for $M > 2$ the iterative method requires more runs. Despite this, it is scalable and requires much less time than the global method, as will be shown later in Section 5.5.2.

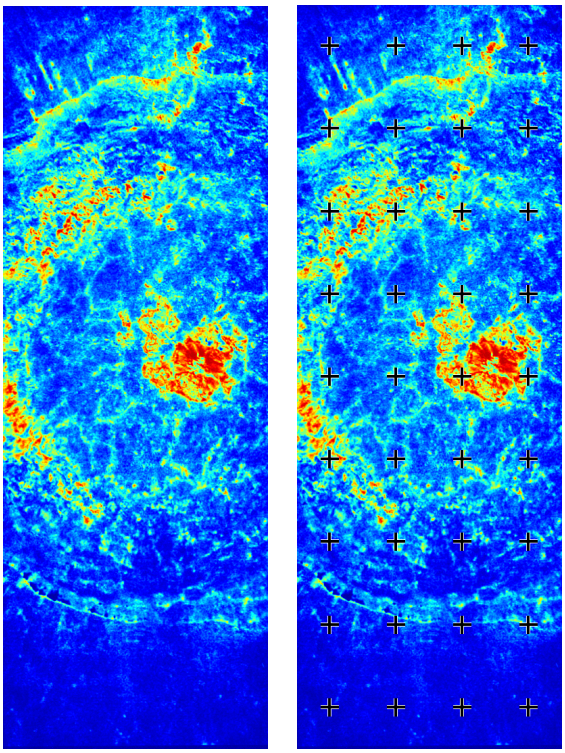
5.5.1. Validation tests

Our first aim was to validate the approaches taken in Section 5.4.1 related to the single cluster coverage computation. Several tests were performed comparing the importance expected by the analysis with the experimental value obtained by averaging random instances of the network using the optimal parameters selected by the solver (*e.g.* see Figure 5.5). Table 5.2 summarizes the comparison between the analysis and the experimental procedure (averaging the result of 5000 experiments). The ratio of importance $\Gamma_{\%}$ is relative to the total importance of the map. These results indicate that the importance captured is derived with less than 4.1% deviation in the worst case, despite the approximations taken in the analysis.

An additional test was performed using the constant importance map $\alpha(x) = 1$ for all $x \in \mathcal{X}$. In this experiment a cluster of N=100 nodes was aimed at the center of coordinates, with a variable dispersion σ ranging from 10 to 100. The importance covered was computed analytically using Eq. (5.14) and experimentally averaging the results of 5000 random deployments. The results are shown in Figure 5.6. The divergence is below 3% in the worst case, and more remarkable is the fact that the optimum was reached for the same dispersion. This shows that, even though there

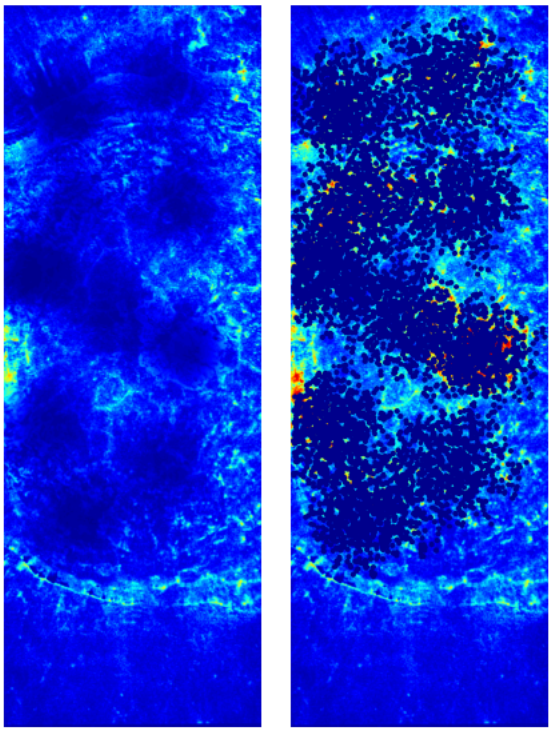


(a)



(b) (c)

Figure 5.4: (a) LROC WAC image of Tycho crater [NASA/GSFC/Arizona State University], (b) Tycho image from UV/visible camera on the Clementine spacecraft showing the location of fresh, mafic material, (c) Initial levels for x_* marked with $+$, iterative optimization



(a) Analytic (b) Experimental

Figure 5.5: Analytic vs experimental multi-deployment Tycho map for M=10, N=800, r_s=4, r_t=12. Dark blue areas indicate the locations covered

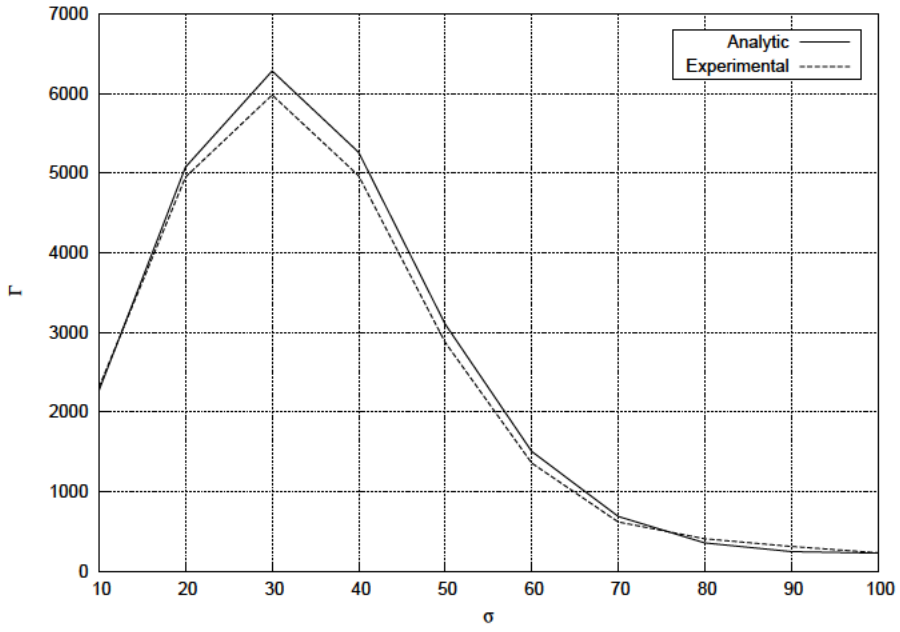
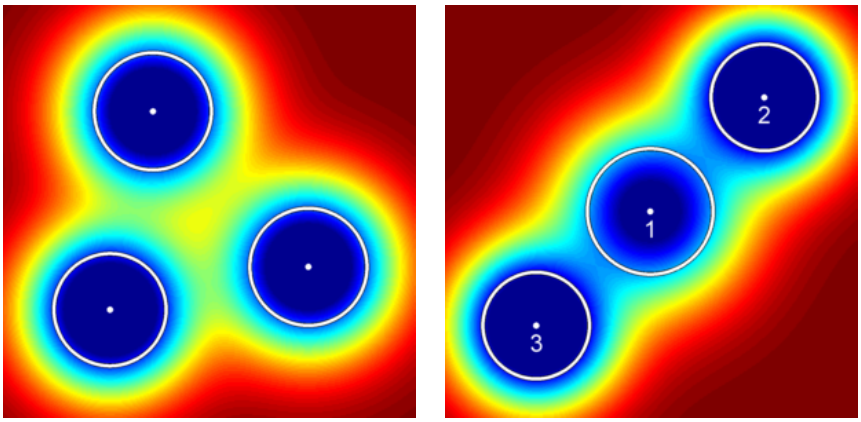


Figure 5.6: Analytic versus experimental importance, homogeneous map, N=100 nodes



(a) Global (b) Iterative

Figure 5.7: Multi-deployment solution, global method, homogeneous map, for M=3, N=600, r_s=4, r_t=12. Red intensities indicate areas not covered, blue ones indicate areas covered.

exists a small divergence in the predicted importance, the parameters computed for the cluster are optimal.

In the next sections all the importances shown were computed experimentally by averaging 5000 samples using the cluster parameters obtained using the optimization procedure; therefore, these values represent the real importances.

5.5.2. Global versus Iterative solution

These tests aimed at comparing the performance of the global and iterative algorithms. As stated in Sections 5.4.2 and 5.4.3, if the size of the map is comparable to the size of the cluster the global optimization algorithm should perform better. This can be checked in the example shown in Figure 5.7. The importance map is a 300×300 area with $\alpha(x) = 1$ for all $x \in \mathcal{X}$. The global solution covers 42.7% of the whole importance, and the iterative one 38.8%. Besides, the first cluster in the iterative case covered 15.3% of the importance, whereas the best cluster in the global case covered only 14.7%. This example shows that it is better to have equilibrated clusters (global case), rather than a good cluster that prevents good placement of the subsequent clusters (iterative case).

However, if the map is larger, this effect is less noticeable. Both optimization strategies were applied to the Tycho map (see Table 5.3 for a summary of the results). Both optimization methods were able to cover the same importance. However, the global method required more computing resources (both CPU time and memory). In fact, for more than 5 clusters the GAMS/CONOPT global implementation was unable to reach a solution due to the lack of computational resources.

	Iterative			Global		
M	Γ(%)	Time (m:s)	Memory (Mb)	Γ(%)	Time (m:s)	Memory (Mb)
1	0.07	08:22	577	0.07	17:02	577
2	0.13	16:44	577	0.13	77:28	1736
3	0.18	25:06	577	0.18	193:18	3526
4	0.22	33:28	577	0.22	369:36	5946
5	0.27	41:50	577	0.27	594:00	8995

Table 5.3: Performance of iterative vs global methods, Tycho map, N=600, r_s=4, r_t=12 (CPU Intel Core i7 2600K)

Summarizing, the iterative method is scalable and achieves similar results to the global method in scenarios in which the target area is much larger than the size of the cluster. Since most real scenarios will be large and require a large number of clusters, the iterative method must be selected. The next sections analyze the performance of this method in depth.

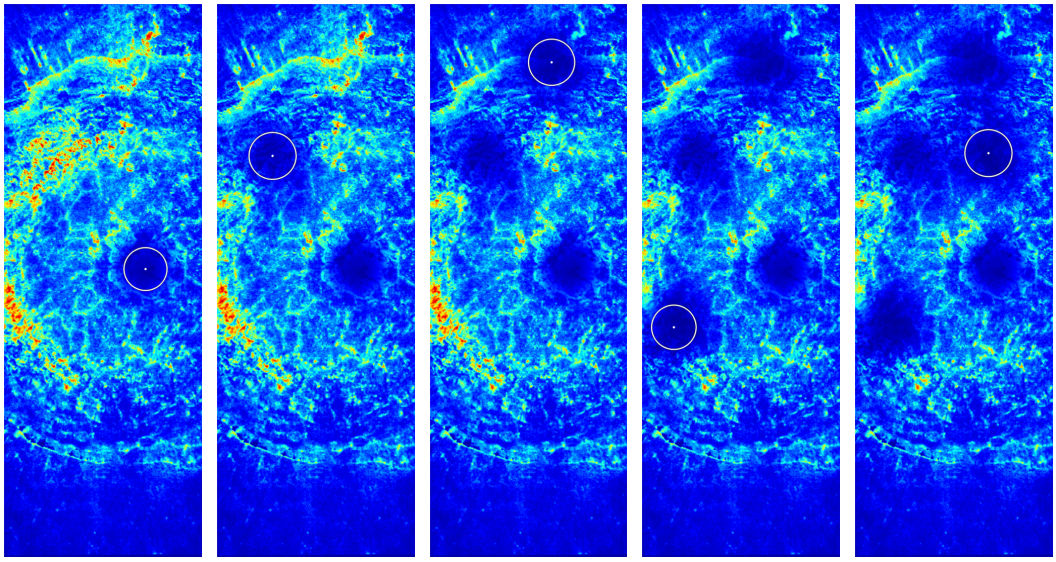
5.5.3. An iterative optimization example: The Tycho crater

In these tests we selected $M = 10$, $N = 800$, and three combinations of sensing/communication ranges: (r_s=2, r_t=6), (r_s=4, r_t=12), and (r_s=6, r_t=18). As an example of the optimization process for the configuration (r_s=6, r_t=20) Figure 5.8 shows one frame per cluster. Frame j represents the importance that remains after the step, *i.e.* $\mathbb{E}\{\alpha^{j+1}(x)\}$. In each frame the point depicts x_* and the dispersion σ is marked with a circle. The solution found with our algorithm covers almost all of the important zones of the map.

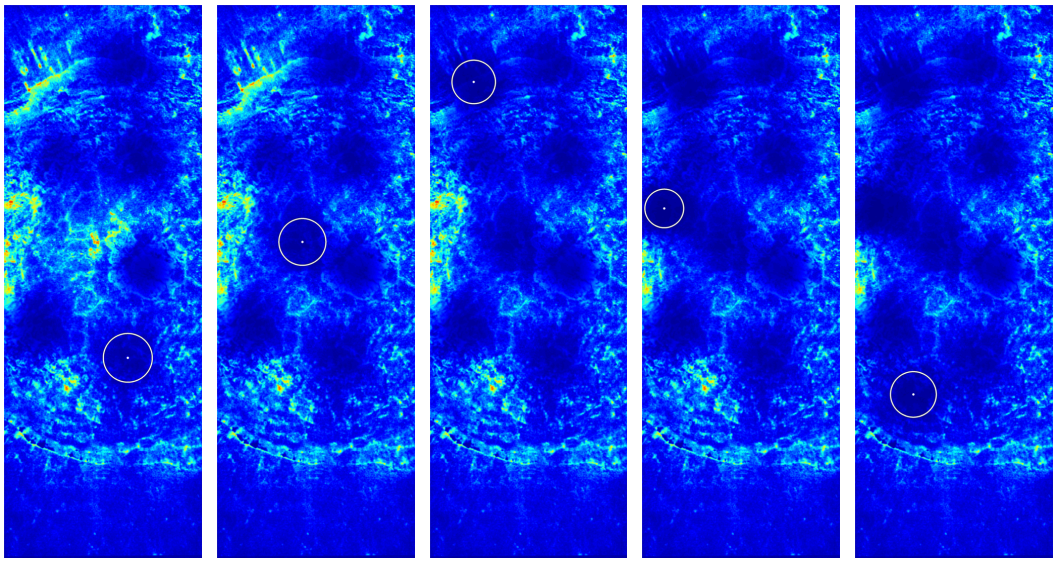
Figure 5.9 shows frames summarizing the optimal solution for each configuration of (r_s, r_t). Besides, the cumulative ratio of importance $\Gamma_{\%}$ per cluster relative to the total importance is shown in Figure 5.10. Note how the configuration of the ranges drastically affects the solution. There is almost no variation in σ for the low ranges whereas there is noticeable variation for mid and high ranges. As can be seen in the figures, the importance gathered per cluster decreases as the steps increase. This is also an expected effect since the first clusters cover the best sites.

In addition, elapsed computation times on an Intel Core i7 2600K were 30 minutes per cluster placement, hence it took only 5 hours for the optimization of a multi-deployment of M=10 clusters. Therefore, the speed of the optimization algorithm is notably fast.

Note that a single cluster captures only a small ratio of the whole region importance (3%, 9%, and 16% depending on the sensing range) for N=800. If the number of nodes is smaller, this effect is more noticeable, *e.g.* for N=100 with r_s=4 (see Figure 5.10) a single cluster collects only 1.82% of the whole importance. Clearly, to cover large areas either a cluster needs to be composed of a very large number

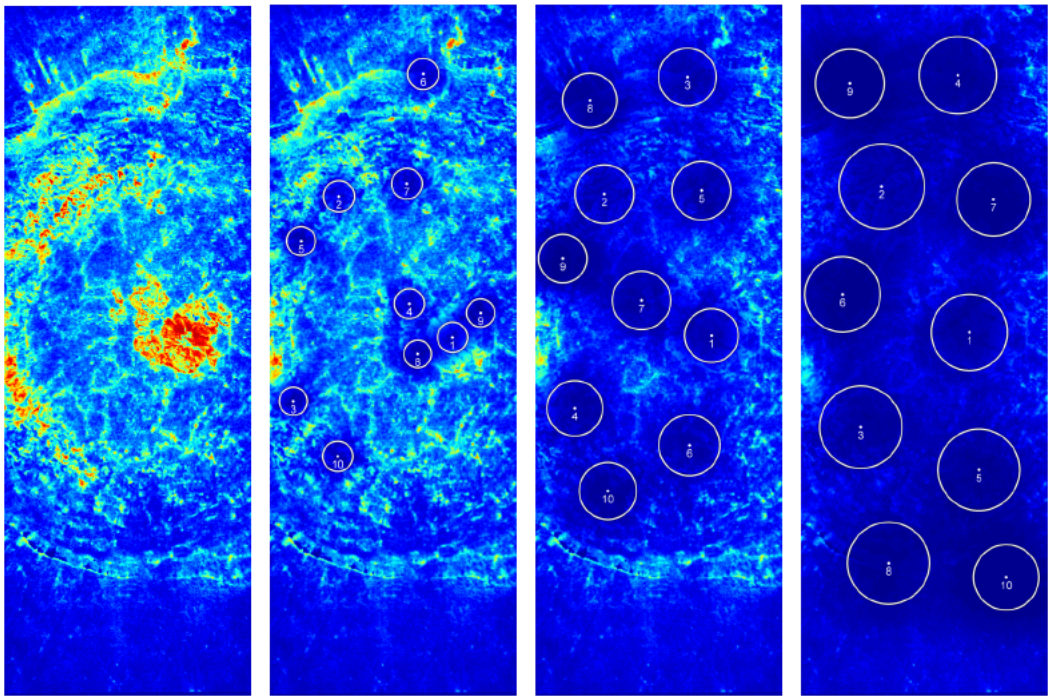


(a) j=1 (b) j=2 (c) j=3 (d) j=4 (e) j=5



(f) j=6 (g) j=7 (h) j=8 (i) j=9 (j) j=10

Figure 5.8: Multi-deployment solution for Tycho map, M=10, N=800, r_s=4, r_t=12



(a) Original (b) $r_s = 2$, $r_t = 6$ (c) $r_s = 4$, $r_t = 12$ (d) $r_s = 6$, $r_t = 18$

Figure 5.9: Final multi-deployment solution for Tycho map, M=10, N=800

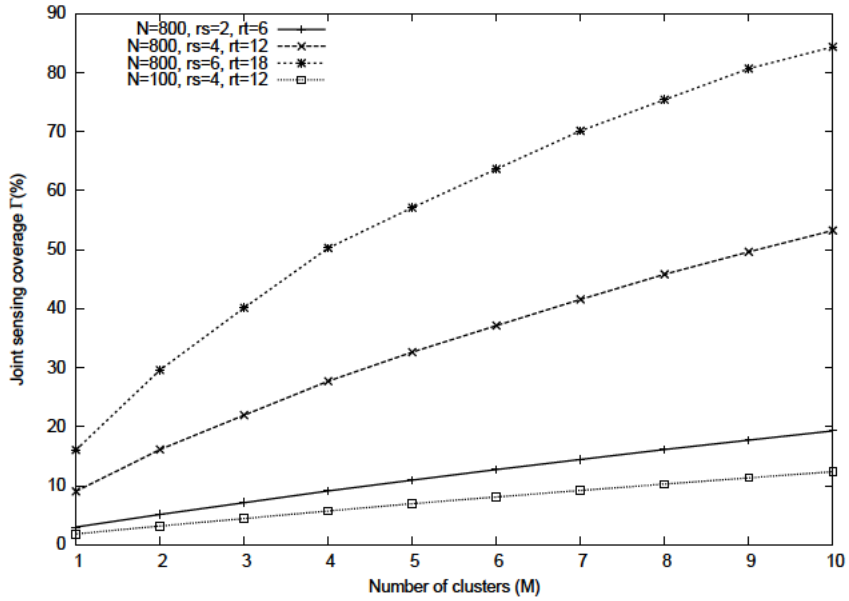


Figure 5.10: Joint sensing coverage $\Gamma(\%)$ versus M, Tycho map

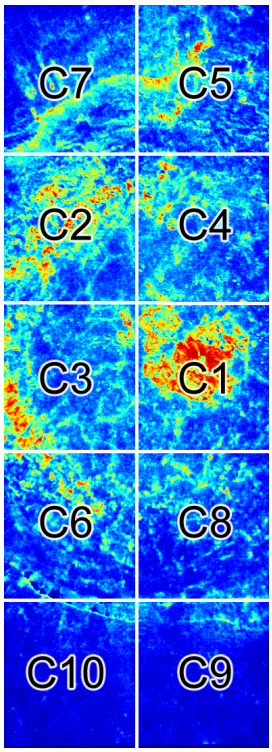


Figure 5.11: Cell division for the heuristic placement, Tycho map

of nodes or several clusters are required. The former approach may be unfeasible if the number of nodes is too high, and, more important, our results indicate that several clusters achieve better sensing coverage than a single cluster with the same number of nodes. For example, with N=800, a single cluster (let us assume r_s=4) can collect 9% of the whole importance, but 8 clusters of N=100 are able to collect 12.35%.

5.5.4. Iterative optimization versus heuristic placement

In the final tests, the optimal strategy computed with our algorithm is compared against a general heuristic based on related works [Leoncini et al., 2009; Zou and Chakrabarty, 2003] which divide the target area into a grid and use one cluster per cell. Following the results in [Leoncini et al., 2009], the optimal target point for the cluster must be the center of the cell. In the implemented heuristic we follow this approach by: (i) dividing the target area into M=10 equally spaced cells (Figure 5.11), (ii) aiming the cluster at the center of the cell and selecting σ values 25, 50, 75, and 100. The solution computed with our optimal approach outperforms the heuristic in all cases. The improvement percentages are reported in Table 5.4.

Even selecting the best σ parameter in the heuristic approach (we must remark that previous works do not provide methods to compute this) our approach improves the result by more than 14% in the worst case (N=800). In a scenario

	σ=25	σ=50	σ=75	σ=100
N=100	**82,59%**	1989,28%	5624,79%	8554,04%
N=200	**47,71%**	369,96%	4116,63%	9345,81%
N=400	**53,76%**	54,42%	643,22%	5626,80%
N=600	67,35%	**24,41%**	132,55%	1680,51%
N=800	77,07%	**14,24%**	56,48%	401,79%

Table 5.4: Improvement in terms of covered importance of the optimal solution compared with the heuristic. Tycho map, M=10, N=800, r_s=4, r_t=12

with fewer nodes, where the optimization problem is more challenging, the optimal approach developed clearly outperforms the heuristic (*e.g.* by more than 82% for N=100). The difference may even greater if σ is poorly selected (*e.g.* more than 400% in all cases if σ=100).

5.6. Conclusions

In this Chapter we have developed mathematical and algorithmic procedures to calculate the best placement options for WSN Gaussian clusters in an area of non-homogeneous importance. In the formulation of the optimization problem we have captured the main characteristics of the WSN model and taken realistic assumptions: the clusters must be normally distributed over the monitored area and must also cover the best areas. At the same time, they must be connected, i.e. nodes cannot be isolated.

We have shown that the approximations selected in our analytical procedure achieve notably good results, closely resembling those obtained in experimental evaluations. Besides, as an important conclusion, in a single cluster optimal deployment hundreds (if not thousands) of sensors are required to completely sense some relative "small" regions. Iterative multi-deployment proves to be an effective strategy to overcome this problem. Results demonstrate how large non-homogeneous regions may be covered with several clusters using suitable configurations. The solution to the optimal placement problem developed in this work provides such configurations and largely outperforms previous heuristics.

Chapter 6

Optimal planning of WSN deployments for in situ lunar surveys

Look behind you!
A three-headed monkey!

6.1. Introduction

Wireless Sensor Networks (WSNs) are a promising tool for in-situ planetary exploration missions. A number of conceptual projects funded by NASA Innovative Advanced Concepts (NIAC) around this concept have been under study, *e.g.* [Laboratory, 2014; Hemmati et al., 2014; Manobianco et al., 2002; Dubowsky et al., 2005]. The common element of these projects is the idea of distributing a high number of small and reduced-cost sensing elements (*nodes*) to monitor directly the chemistry and geology over a large surface. Data is conveyed to special (*sink*) nodes for transmission and processing by a communication network formed by same nodes. Several works, e.g. [Prasad and Murty, 2011; Dubois et al., 2009; McCleese et al., 2001; Gaura and Newman, 2006; Dubowsky et al., 2005], have discussed the expected advantages of this kind of mission architectures in contrast to other in-situ alternatives, such as landers or rovers:

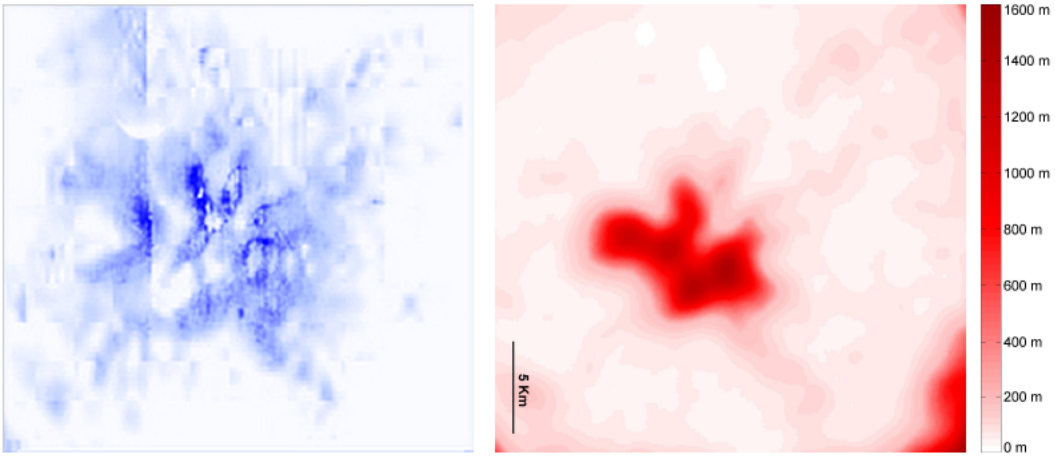
- Exponential increment of the spatial and temporal resolution achievable.
- Large observational range.
- Study of rough regions, inaccessible by other means.
- Reduced cost.
- Robustness for acquiring valid data in harsh environments due to the redundancy of the elements.
- Unique observational capabilities based on the availability of several concurrent observation points, e.g. seismology studies.

As long as these proposals mature, several aspects shall be addressed. Among them, a critical challenge is deployment planning [Younis and Akkaya, 2008], which requires consideration about the several objectives involved (like in any exploration mission), such as terrain interest, energy consumption, sensing coverage, network lifetime, connectivity and so forth. Often these objectives are conflicting and operational trade-offs must be necessarily established during deployment design. The goal of this work is developing an analytical model that brings together these aspects and their interactions, allowing to find optimal launch configurations, and to analyze the trade-offs among the different mission parameters.

The main characteristic of the network deployment in this kind of missions is that node placement is inherently random due to airborne sensor launch, forming one or more groups or *clusters* (one per launch). Due to this, connectivity and sensing properties of the WSN are also random, and the deployment design must face this issue by selecting suitable launch parameters (e.g. launch position, launch height, etc.) that guarantee good observational capabilities. This design is also affected by the heterogeneous relevance of the region, where some zones must be monitored with higher priority to foster potential for scientific discoveries. Moreover, two basic constraints must be imposed in the deployment planning: (i) a mass limitation for the overall mission (nodes, launch equipments, propellants, etc.), and (ii) a minimal network operational time.

6.1.1. Mission background

Although the approach followed in this work can be generalized to different deployment scenarios, we have focused on a particular survey mission to remark the required considerations of a real WSN survey. Namely, we selected the characterization of a WSN deployment to study water concentration over the Moon's Bullialdus crater. Interest for this type of missions has been fostered by recent lunar orbiter missions such as the Chandrayaan-1[Datta and Chakravarty, 2004], which has provided definitive evidence for the presence of various forms of water over the



(a) OH^- absorption strength grades white (0%) to blue (4%) in the central peak of Bullialdus Crater

(b) Digital Elevation Model (DEM) of the central peak of Bullialdus Crater. Source: NASA/GSFC/Arizona State University

Figure 6.1: Bullialdus Crater

lunar surface, and by studies like [McCleese et al., 2001] which have stressed the need of in situ exploration of specific areas suspected of harboring water.

Our target scenario is based on the analysis, provided by [R. Klima and Lawrence, 2013], of the spectroscopic data from the Moon Mineralogy Mapper (M3) instrument of the Chandrayaan orbiter, where remote detection of magmatic water in Bullialdus Crater on the Moon is discussed. Unlike previous studies [Pieters et al., 2009; Sunshine et al., 2009; McCord et al., 2011] that suggest OH^- and H_2O molecules are produced in situ through the interaction of solar-wind-derived protons with lunar soil, [R. Klima and Lawrence, 2013] proposes that hydroxyl absorption features in Bullialdus Crater are inconsistent with a superficial origin, and they are bounded to magmatic mineral that was excavated from depth by the impact that formed the crater. Authors provide an OH^- absorption map of the central peak of Bullialdus Crater (see Fig. 6.1), showing that it is significantly enhanced in hydroxyl relative to its surroundings. These maps provides a direct indication of the relevance of each exploration zone.

The goal of in-situ exploration would be to determine actual water concentrations. A number of strategies could be used to characterize the soil from in situ measurements, e.g. see the work by [Buehler et al., 2005]. Although the previous work concentrates specifically on lunar soil of silicates and oxides, specific methods to determine water concentration are also possible based on soil conductivity and dielectric constant measurements as reported by the same team in [Buehler et al., 2006]. According to them, sensing equipment can be miniaturized and requires low power, thus it may adapt well to the hardware capabilities of a WSN. Other works have proposed similar concepts based on impedance based sensors [Pabari et

al., 2013], or dielectric spectroscopy [Nurge, 2012] targeted to detect ice water in permanent dark regions (not the case of Bullialdus crater).

6.1.2. Work contributions

The main contributions of this work are:

- As far as authors know, previous WSN deployment optimization models considering effects of 3D terrain in connectivity and the heterogeneous coverage relevance do not exists.
- Similarly, setting the main model constraint as a mass limitation is also a novel approach of our work. Mass computations include the hardware and the propellant consumed to reach the clusters launch positions. Our model describes the spacecrafts maneuvers required and derives the propellant requirements by means of Tsiolkovsky-rocket equation.
- The number of clusters and nodes per each cluster are variables that the optimization algorithm freely decides within the mass limitation, unlike previous models.
- Nodes positions are linked to real deployment variables such as the launch height of the cluster, instead of merely considering the abstract node distribution parameters.
- The model describes the relationships among network lifetime, masses, sensor placement mechanism, node radio hardware, etc. The main trade-offs observed in the optimal solutions are also analyzed.

In the rest of this work we discuss related works in Section 6.2, the mission architecture is introduced in Section 6.3, while the next sections describe different aspects of our analytical model, leading to the optimization problem summarized in Section 6.7 which is addressed by means of a simulated annealing (SA) in Section 6.8. Results and analysis of trade-offs are carried out in Section 6.9, followed by the conclusions.

6.2. Related works

Deployment optimization with a random placement component has an increasingly interest due to the forecast applications of WSN and more general wireless mesh networks.

A number of works such as [E.N., 1961; Ammari and Das, 2008; Isler et al., 2004; Liu and Towsley, 2004] focus on the topology properties of ad-hoc networks. [E.N., 1961] studied the critical density of a Poisson Process node distribution at which a

network can provide long-distance multi-hop communications. [Ammari and Das, 2008] analyzed the degree of sensing coverage and connectivity of a WSN, assuming that sensors are deployed following a Poisson point distribution, establishing a relationship between coverage and connectivity for this kind of deployment. [Isler et al., 2004] investigate the density of nodes required to guarantee coverage and connectivity in an uniform deployment of sensors. [Liu and Towsley, 2004] characterize, by means of simulations, the fundamental coverage properties of a WSN. All these works aim at selecting basic network properties such as node density, communication range, and so forth, for *non-clustered* networks. In contrast, we take into account the non-homogeneous communication properties and clustered nature that result in a real deployment.

Some works have studied clustered deployments, considering mixed connectivity-coverage objectives, similarly to us. [Wang et al., 2008] iintroduce two algorithms for Gaussian deployments which reduce the number of sensors while satisfying the required coverage and lifetime conditions, although their model can only be applied to small scenarios with a high density of nodes. [Sevgi and Koçyigit, 2014] address the issue of *partial connected coverage*, a composite connectivity-coverage metric, leading to deployments where some degree of coverage and/or connectivity may be sacrificed when sensor density is limited. On the contrary, our model is constrained by mass and lifetime limitations that arise in planetary exploration.

Hybrid [Hayajneh and Khasawneh, 2014] and two-tier deployment strategies [Bari et al., 2008; Xu et al., 2010; Bari et al., 2011] have also been analyzed and result in better deployments in terms of coverage and other metrics. However, these works do not bound their theoretical distribution to a realistic airborne deployment, as we do in this work.

However, the aforementioned works do not consider two relevant aspects of real deployments that are central to our work: heterogeneous coverage relevance in the target area, and realistic connectivity models considering the impact of 3D surfaces. Some works deal with the first aspect, but as far as the authors know, contributions to the second aspect barely exist.

On the one hand, some works consider the heterogeneous coverage relevance. [Huang and Tseng, 2005] mention this issue and propose an algorithm for covering important areas with more sensors. [Dhillon et al., 2002] discuss the inherent uncertainty of sensor readings and assign a probability measure to sensor detections. [Aitsaadi et al., 2010] introduce a multi-objective deployment algorithm (MODA) which takes into consideration deployment cost (number of sensors), event detection probability, connectivity, and energy consumption. The proposed algorithm minimizes energy consumption and the number of sensors required to accomplish coverage goal. Similarly to our work, their deployment scenario has heterogeneous sensing requirements. [Dasgupta et al., 2003] propose a scenario in which a certain amount of data has to be gathered by the network. The data is non-uniformly

distributed and the authors propose a heuristic that relocates nodes in order to balance the load of data transmitted from zones with greater density of data. [Isler et al., 2004] models coverage with a weighting function, which assigns a reward to each point covered that is inversely proportional to the distance from the sensor. Sensor positions that are likely to increase the total weight are preferred. [Krause et al., 2008] propose a combinatorial optimization problem to maximize mutual information obtained by the WSN. [Wang et al., 2006] present an optimization model to identify the sensor locations that maximize the average probability of event detection per point, but they restrict the candidate set of node positions. In [Vales-Alonso et al., 2013b] we developed an analytical model that aims at maximizing the joint sensing coverage of the network, by selecting suitable Gaussian cluster positions and dispersions.

On the other hand, [Jin et al., 2012] tackle the problem of sensor deployment over 3D surfaces, aiming to achieve the highest overall sensing quality. Based on conformal mapping and under a generalized centroidal Voronoi partition, authors propose a series of algorithms to compute the optimal sensor deployment on 3D surfaces (topologically equivalent to a disk) that minimizes the overall sensing unreliability of the WSN. However, this work is focused exclusively on sensing quality, while network connectivity is barely addressed. [Temel et al., 2014] propose a deterministic sensor deployment method based on wavelet transform and solved by means of a Cat Swarm Optimization algorithm, that aims to maximize the quality of coverage of a WSN while deploying a minimum number of sensors on a 3D surface. As [Jin et al., 2012] did, network connectivity issues are hardly taken into account the network.

Finally, in relation to WSN Lunar surveys, [Prasad and Murty, 2011] also present a sensor node prototype for water detection and propose two deployment schemes over the Shackleton Crater: a first scheme that uses an orbital deployment and a second that utilizes a projectile deployment from a Rover or Lander. Authors have carried out simulations for both scenarios to estimate the number of nodes required and the communication range between nodes for covering the target crater. Orbital deployment scheme allows to cover a large area with a random deployment (assuming a normal Gaussian distribution). Unlike our work, they do not aim at searching for optimal deployment planning, but presenting some simulated results that endorse their proposal.

Summarizing, several previous works have analyzed WSN random deployments, but in all cases with simplified connectivity models and without modelling the real constraints to which a real planning is faced in planetary surveys.

6.3. Lunar mission concept

In this section we define the characteristics and stages of the envisioned lunar survey mission. Moreover, we introduce the variables, main components and the conceptual goal of our problem.

We consider a deployment area $\mathcal{X} \subset \mathbb{R}^2$, where an *importance function* $\alpha : \mathcal{X} \to \mathbb{R}_0^+$ and an *elevation map* $z : \mathcal{X} \to \mathbb{R}_0^+$ are defined. The former provides information about the plausibility of obtaining suitable information about the variable of interest at a given location in the terrain. This quality map can be obtained from previous mappings of this variable or from another one correlated with it. Besides, the latter provides the terrain height from a reference ground, which is needed for computing accurate radio propagation among nodes.

The survey mission aims at deploying over this surface c clusters with $n_1, n_2, \ldots, n_c$ nodes each, in order to maximize the importance gathered by the network. The deployment stages considered in this mission are shown in Fig. 6.1 and are the following:

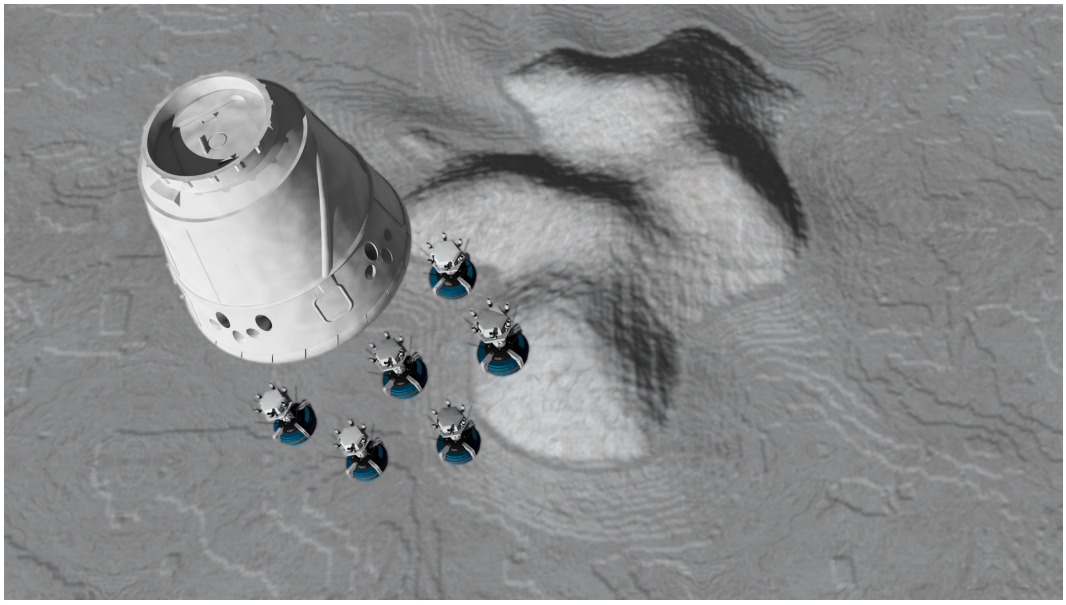
1. *Probes separation.* Deployment probes are detached from the main spacecraft from a position called *insertion point* (x_0, h_0). Each deployment probe must reach its assigned *deployment point* (x_i, h_i) for i=1, ..., c from which it will launch the corresponding cluster of sensors.

2. *Clusters deployment.* From each designated deployment point, the nodes of the ith cluster are released over the target area. Nodes position are considered independent and identically distributed (iid) random variables, whose distribution can be controlled. For example, the nodes dispersion could be related to the release altitude h_i or other adjustable parameters in the deployment mechanism. Let us denote position of the jth node of cluster ith as $\boldsymbol{x}_i^j \in \mathcal{X}$.

3. *Probes landing.* Following nodes release, each deployment probe lands directly on the vertical below the deployment point, i.e. in the *landing point* $(x_i, z(x_i))$.

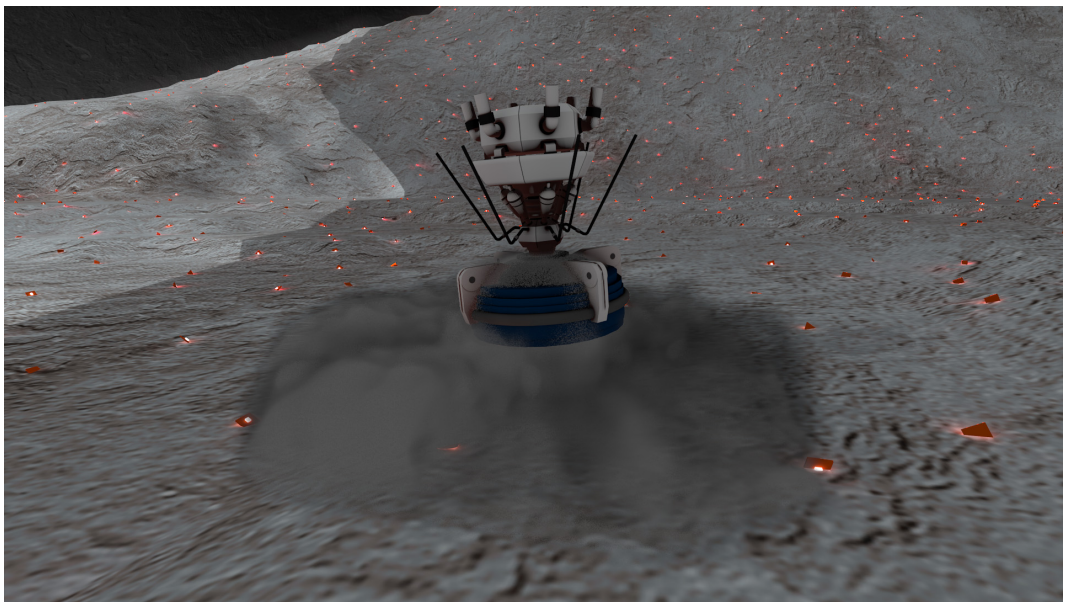
4. *Operation.* After landing, the deployment probe acts as the sink of the WSN, receiving sensed data from connected nodes and forwarding them to Earth-based control centers for processing and analysis. To reduce interferences, we assume each cluster operates independently, communicating in different spectrum bands. Nodes can connect to the sink either by a direct radio link, or via a multi-hop communication through other nodes in the cluster, when such a path exists.

Summarizing, the goal is establishing proper deployment points and launch parameters for each cluster in order to maximize the importance gathered. It will

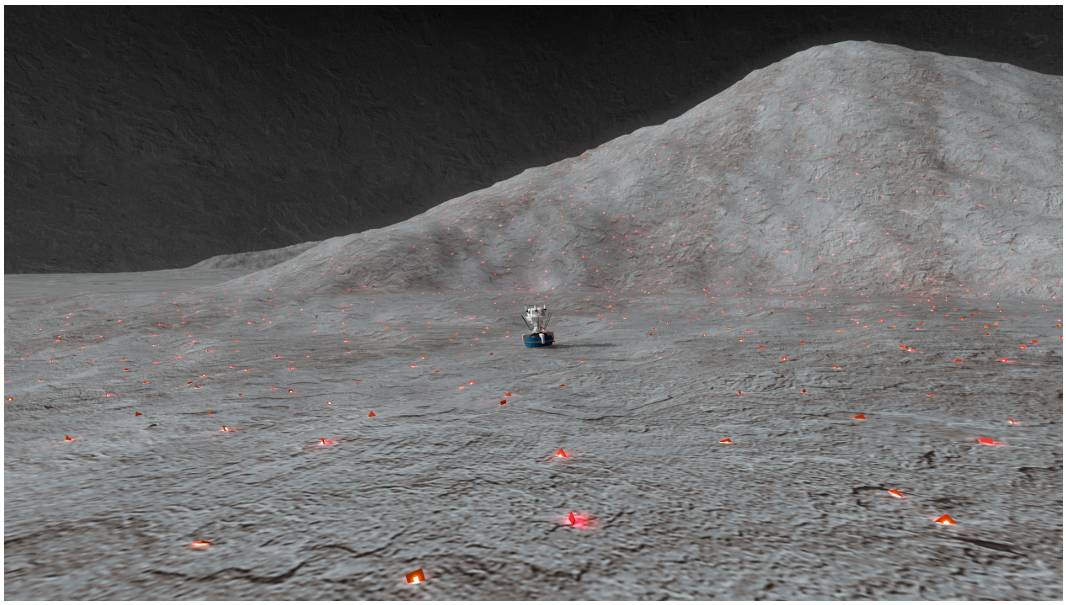


(1) Probes deployment

(2) Clusters deployment



(3) Probes Landing



(4) Operation

Figure 6.1: Overall sensors deployment strategy

depend on the connectivity and sensing capacity of the WSN, which is studied in depth in next Section 6.4. The essential characteristic of this kind of deployment is that nodes position is random and therefore both the connectivity and the sensed information will be random as well. This randomness shall be incorporated in the model, and suitable objective functions must be established. This is also done in Section 6.4.

In the problem definition, in addition to maximize the information captured, two critical limits are faced in the deployment design. First, a maximum payload P is available to the mission. This includes the mass of the nodes and the mass of all deployment probes (one per cluster) and their propellants, required to reach the designated deployment points. On the other hand, a minimum mission lifetime T must be guaranteed. Sections 6.5 and 6.6 deal with each aspect, respectively.

Once the proper models are described, our problem is stated formally in Section 6.7 as a Mixed-Integer Non-linear Non-Convex mathematical program, which is solved by means of simulated-annealing meta-heuristic described in Section 6.8.

6.4. Sensing-connectivity model

We assume all nodes are homogeneous and when placed at point $(x, z(x)) \in \mathcal{X} \times \mathbb{R}_0^+$ they are characterizing the variable of interest within an open ball $B(x, r_s) \subset \mathcal{X}$. This can be interpreted either as the capacity of obtaining direct measurements in this area, or, more usually, that the point measurement at x offers correlated information of points nearby, that is, those within the open ball $B(x, r_s)$. In the latter case, r_s should be interpreted as the maximum range until which correlated information can be obtained, while for the former r_s has the straightforward meaning of sensing range. Thus, the importance covered by a sensor placed at x is:

$$\int_{B(x,r_s)} \alpha(x) \mathrm{d}x \tag{6.1}$$

Due to the random node scattering it is possible that some nodes are isolated in the WSN. That is, that no direct or multi-hope route exist to the sink. Only nodes connected, *active* nodes henceforth, are able to convey information. Moreover, if the sensing areas of active nodes overlap, we assume that importance captured in those areas does not increase. This models the fact that sensing replicas are not offering new information about the variable of interest. Hereafter, let us denote the set of active nodes in cluster i as Φ_i. Therefore, the total importance capture, or *network coverage* Γ, is given by:

$$\Gamma = \int_{\mathcal{B}} \alpha(x) \mathrm{d}x \tag{6.2}$$

where $\mathcal{B}$ is

$$\mathcal{B} = \left[\bigcup_{i=1}^{c} \bigcup_{j \in \Phi_i} B(\boldsymbol{x}_i^j, r_s) \right] \bigcap \mathcal{X} \tag{6.3}$$

Since node position is random, $\mathcal{B}$ is random as well. To deal with this randomness we propose to establish as the target goal of our problem the maximization of the expected value of Γ, $\mathbb{E}\{\Gamma\}$. Other criteria, such as the maximization of the 95th or 99th percentile value or mixed expectation-variance models could be established. Although the latests are not directly analyzed in this work, they could be formulated in a similar basis.

To connect this sensing-connectivity model with the problem variables described in previous section is necessary to relate the deployment parameters to the node positions, and them to the network connectivity. This is addressed in the next two sections, respectively.

6.4.1. Sensor ejection mechanism

Sensors are launched from the deployment probe by means of an ejection mechanism where a gas expels the cargo. Figure 6.2 sketches the parameters of the ejection mechanism. The deployment probe must acquire a release velocity $\boldsymbol{v}_r$ for proper operation. For simplification we assume $\boldsymbol{v}_r$=0 in this work, although models can be straightforwardly modified to consider a different velocity. We consider that nodes are ejected from the probe at a nominal speed of $v_s > 0$ and with an angle less than θ (see Figure 6.2), where $0 < \theta < \pi/2$. Then, nodes fall with parabolic motion from the release height above surface $h_i^* = h_i - z(x_i)$. Hence the nominal maximum distance at which nodes land from the landing point $(x_i, z(x_i))$ is given by,

$$\delta_i = \frac{\sin^2(\theta)(\sqrt{v_s^2(v_s^2 \cot^2(\theta) - 2g_0 h^* \csc^2(\theta))} + v_s^2 \cot(\theta))}{g_0} \tag{6.4}$$

where we assumed a constant surface height of $z(x_i)$ around the deployment point, and where g_0 = -1.6249 m/s^2 is the gravitational acceleration constant in the Moon.

The previous formula provides the nominal maximum landing distance, but several other factors may determine the real position. For example, in practice most nodes will be ejected with less angle than the maximum θ and will be closer to the landing point. Besides, many other effects may occur such as nodes hitting each other during the launch, terrain height is prone to have variations causing differences in the landing distance, and so on. Moreover, the expulsion mechanism is supposed to has rotational invariance, that is, the angle of the position with the

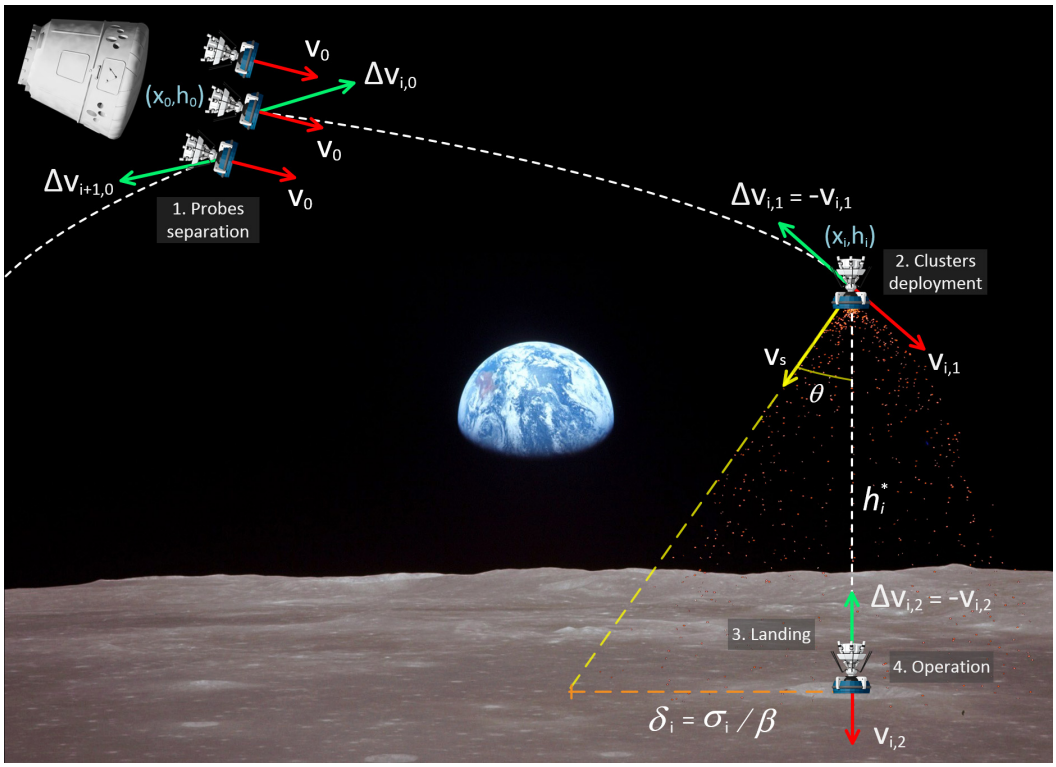


Figure 6.2: Deployment sequence

landing point is assumed to be random, uniformly distributed and independent of the landing distance.

A suitable random distribution that displays the previous conditions is a Gaussian one. As it is known, it arises in physical process where many different independent effects combine and the density function has the rotationally invariant property, such us in our case. Besides, they correspond to actual observed distributions in airborne seeding [George et al., 2004; Zou and Chakrabarty, 2003; Fang et al., 2006]. Thus, we have considered that node positions in the ith cluster are given by iid normal bivariate random variables $N(x_i, \sigma_i I)$, where $\sigma_i = \beta\delta_i$ being β a launch parameter that must be suited to the particular ejection mechanism.

6.4.2. Cluster connectivity

WSN nodes usually operate in the Industrial Scientific Medical (ISM) Ultra High Frequency (UHF) band of 2.4 GHz with a transmission power between 0 dBm and 20 dBm. Although with this transmission power communication ranges in the order of a few hundreds of meters are possible with direct line of sight, in practice communication range can be remarkably lower, and it is never homogeneous for all node pairs. Real (3D) terrain profoundly impacts the connectivity, which must

Parameter	Value	Description
WSN Model		
SNIR_{min}	10 dB	Signal to Noise plus Interference Ratio threshold
G_t	2.14 dBi	Gain of transmitter antenna
G_r	2.14 dBi	Gain of receiver antenna
c_0	$3{\times}10^8$ m/s	Speed of light in vacuum
f	2.4 Ghz	Operating frequency of the sensor
B	100 Khz	Signal bandwidth
N_0	-122.68 dBm	Background Noise at 390K (worst case)
F	3 dB	Noise figure
σ_h	220.93 m	RMS surface height
Deployment		
γ	2	Structural mass coefficient
g_0	-1.6249 m/s^2	Acceleration due to gravity on the Moon
v_e	162 m/s	Effective exhaust velocity
v_s	20 m/s	Sensors launch speed
θ	$\pi/3$ rad	Nominal parabolic sensors launch angle
$v_{i,0}$	$(10, 10, -10)$ m/s	Initial descent velocity of the sky crane i
Node Hardware		
m_{hardware}	30 g	Hardware mass (physical sensor, electronics, ...)
m_{battery}	See Table 6.3	Battery mass
V	3.65 V	Open circuit voltage of the node
I_{heater}	50 mA *(active)*	Heater current drain during *active* phase
I_{sensor}	20 mA *(active)*	Sensor current drain during *active* phase
I_{sensor}	$5{\times}10^{-4}$ mA *(sleep)*	Sensor current drain during *sleep* phase
I_{radio} (0 dBm)	13.8 mA *(active)*	Radio current drain during *active* phase (P_{tx} = 0 dBm)
I_{radio} (-20 dBm)	7.06 mA *(active)*	Radio current drain during *active* phase (P_{tx} = -20 dBm)
I_{radio} (0, -20 dBm)	$8.22{\times}10^{-4}$ mA *(sleep)*	Radio current drain during *sleep* phase (P_{tx} = 0, -20 dBm)
Simulated Annealing		
InitTemp	100	Initial temperature
StopTemp	10^{-32}	Stop temperature
MaxConsRej	1500	Maximum number of consecutive rejections
MaxSuccess	25	Maximum number of successes within one temperature
MaxTries	2000	Maximum number of tries within one temperature
Generation Rules		
c_{max}	10	Maximum number of clusters
n_{max}	1000	Maximum number of nodes per cluster
Δn	25	Number of nodes quantum
σ_{min}	400 m	Minimal cluster dispersion
σ_{max}	1600 m	Maximal cluster dispersion
σ_0	200 m	Generation of random samples associated to (x_0, h_0)

Table 6.1: Parameters

be duly computed depending on the transmitter and receiver position. Moreover, transmission range depends also on the receiver interference power, which must be taken into account in a realistic model. Given the Signal-to-Noise-Interference ratio, SNIR, communication can take place if it is above a minimum SNIR_{min} (see Table 6.1), which is a radio-hardware dependent parameter. SNIR for the ith cluster, transmitter jth and receiver j$'th$ can be expressed in dBs as:

$$\text{SNIR}_i^{jj'}[\text{dB}] = P_{\text{tx}}[\text{dBm}] - L_i^{jj'}[\text{dB}] - (I_i^{j'} + N_0 + F)[\text{dBm}] \tag{6.5}$$

where P_{tx} is the transmission power in dBm, $L_i^{jj'}$ is the propagation loss in dB and $I_i^{j'} + N_0 + F$ is the interfering power in the receiver plus the thermal noise and the noise figure, also expressed in dBm. The thermal noise is $N_0{=}KT_aB$, where K is the Bolztmann constant, B is the signal bandwidth (see Table 6.1), and T_a is the temperature. Assuming the worst case, daylight temperature, $T_{\text{max}} = 396$ K. The noise figure F is a radio-hardware dependent parameter (see Table 6.1).

Interferences depend on the propagations losses $L_i^{jj'}$, which must be computed with an specific lunar radio propagation model, suited for the spectrum bands assumed in WSN, like the one proposed in [Pabari et al., 2010]. In case of lunar WSNs, sensor nodes are deployed over the surface with very short antenna heights, therefore ground reflected signal is not expected at the receiver. Besides, the lunar surface is a very poor reflector of radio waves. For a radio wavelength $\lambda{=}12.5$ cm (2.4 GHz band), the reflection coefficients, are lower than 0.05 [Krupenio, 1971]. Thus, the model in [Pabari et al., 2010] can be simplified to:

$$L_i^{jj'} = \frac{G_t G_r \lambda^2}{16\pi^2} \left| D_i^{jj'} \frac{1}{d_i^{jj'}} \right|^2 \tag{6.6}$$

where $L_i^{jj'}$ is expressed in natural units, G_t and G_r are the gains of the transmitter and receiver antennas respectively, $D_i^{jj'}$ is the diffraction loss for the direct path (i.e. line of sight signal) and $d_i^{jj'}$ is this direct-path distance between nodes j and j' of the ith cluster. $D_i^{jj'}$ computation method is provided in [Wong, 2002], and it is related to the surface height profile between the transmitter and the receiver, with the main obstacle between them (if any) determining the diffraction loss. Therefore, propagation losses must be computed tailored to the particular transmitter and receiver positions.

The interference power in the j$'th$ receiver depends exclusively on nodes of the same cluster, since clusters operate in separate frequency channels. Moreover, not all nodes cause interferences. In wireless networks, simultaneous transmissions are controlled by the Medium Access Control (MAC) protocol. Therefore, the interference comes from non-direct neighbours, including non-active ones since they will be trying to synchronize with the cluster. Henceforth, let us denote $\Omega_i^{j'}$ the set of non-neighbour nodes of node j$'th$ in cluster ith. Furthermore, nodes usually

Algorithm 1 Adjacency matrix $\boldsymbol{A}_i$ calculation

1: $\boldsymbol{A}_i \leftarrow 1$
2: $P_{\mathrm{rx}_i}^{jj'} \leftarrow P_{\mathrm{tx}} L_i^{jj'}$
3: **while** TRUE **do**
4: $\quad \hat{\boldsymbol{A}}_i \leftarrow$ **not** $\boldsymbol{A}_i$
5: $\quad I_i^{j'} \leftarrow \rho \sum_j P_{\mathrm{rx}_i}^{jj'} \hat{a}_i^{jj'}$
6: $\quad \mathrm{SINR}_i^{jj'} \leftarrow \frac{P_{\mathrm{rx}_i}^{jj'}}{I_i^{j}+N_0+F}$
7: $\quad \boldsymbol{A}_{\mathrm{aux}} \leftarrow \mathrm{SINR}_i^{jj'} \geq \mathrm{SINR}_{\mathrm{min}}$
8: $\quad$ **if** $\boldsymbol{A}_i == \boldsymbol{A}_{\mathrm{aux}}$ **then**
9: $\quad\quad$ **return** $\boldsymbol{A}_i$
10: $\quad$ **else**
11: $\quad\quad \boldsymbol{A}_i \leftarrow \boldsymbol{A}_{\mathrm{aux}}$
12: $\quad$ **end if**
13: **end while**

operate in an active/sleep cycle (*duty* cycle) to save energy. Radio communications and sensing take place only during the active phase. Let us denote as ρ the ratio of the active and sleep period durations. Thus, the probability that a node is in the active phase of the cycle is ρ. In the worst case, assuming that nodes always transmit in their active phase, the average interfering power (in natural units) on the receiver j′th is given by:

$$I_i^{j'} = \rho \sum_{j \in \Omega_i^{j'}} P_{\mathrm{tx}} L_i^{jj'} \tag{6.7}$$

Since interference power depends on the propagation losses it must be also computed tailored to particular positions of *all* nodes in the cluster.

From the previous set of formulas we are able, for a given realization of cluster node positions, to compute the SNIR associated to each pair. Those pairs with SNIR above minimal threshold would be direct neighbours, and those below the minimal SNIR threshold, $\mathrm{SNIR}_{\mathrm{min}}$, would determine to the non-neighbour sets $\Omega_i^{j'}$. Apparently, this is contradictory since these sets must be known *prior* to the computation of the SNIR. What we actually have is an expression of the form $\mathrm{SNIR}_i^{jj'} = f(\mathrm{SNIR}_i^{jj'})$, which can be solved using the Banach's fixed-point method: setting an arbitrary initial SNIR_0 we compute $\mathrm{SNIR}_{k+1} = f(\mathrm{SNIR}_k)$. Eventually, the sequence SNIR_k converges to a system solution.

Algorithm 1 provides an alternative fast implementation to compute the neighbor sets in the network. It operates defining for cluster i the network adjacency matrix, $\boldsymbol{A}_i$, whose element $a_i^{jj'}$ has value 1 if $\mathrm{SNIR}_i^{jj'} > \mathrm{SNIR}_{\mathrm{min}}$ and 0 otherwise. Besides, note in line 5 of the algorithm that only interferences from non-neighbor nodes are considered, since if nodes j and j' are not neighbors, then $\hat{a}_i^{jj'} = 1$, and the interference from j' is summed. In all of our tests the algorithm converged in

a small number of steps, usually in the order of 10.

Once the neighbor sets are computed using Algorithm 1, the connected component with the cluster's sink (direct or via multi-hop route) can be easily found by means of well-known algorithms such as Dijkstra or Tarjan, determining the active set of the cluster, Φ_i.

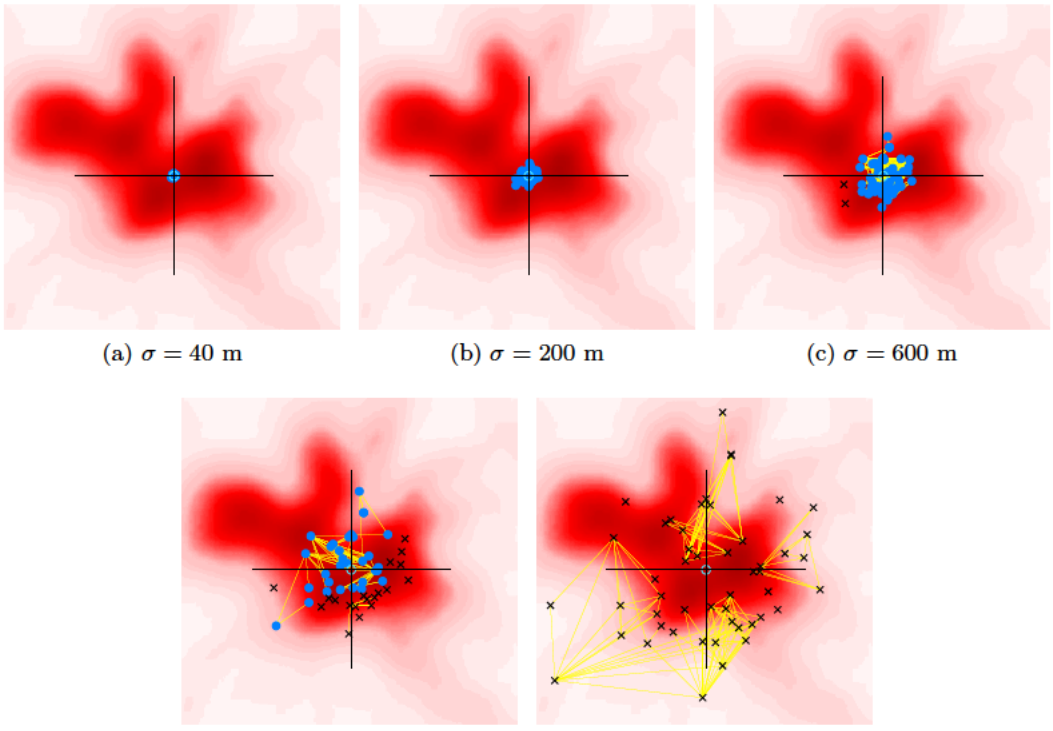
6.4.3. $\mathbb{E}\{\Gamma\}$ computation

Since connectivity depends on the particular realization of the node positions, the network coverage Γ will depend on them as well. Its mean, $\mathbb{E}\{\Gamma\}$, can be approximated numerically by Monte Carlo evaluation, averaging the coverage of random independent realizations until its mean converges.

Summarizing, the Monte Carlo evaluation comprises the following steps:

1. Fix the launch variables: c and x_i, h_i for $i = 1, \ldots, c$
2. For each cluster i compute $\sigma_i = \beta\delta_i$ where δ_i is given by Eq. (6.4).
3. For each cluster i:
   a) Select n_i random node positions, $x_i^1, x_i^2, \ldots, x_i^{n_i}$ with normal bivariate distribution $N(x_i, \sigma_i I)$.
   b) Determine the direct communication links, i.e. those where $\mathrm{SNIR}_i^{jj'} > \mathrm{SNIR}_{\min}$, using the fixed-point method.
   c) Determine the active nodes set Φ_i using Tarjan's algorithm.
4. Determine the set $\mathcal{B}$ given by Eq. (6.3).
5. Compute the realization of Γ by evaluating Eq. (6.2).
6. Repeat from 3) until $\bar{\Gamma}$ converges.
7. Return $\mathbb{E}\{\Gamma\} \leftarrow \bar{\Gamma}$

This procedure is computing-intensive, since for each Γ sample all the positions and cluster-dependent computations must be reevaluated, and then the fixed-point method must be run until convergence. In Section 6.8 a method to speed up this computation is described, it is based on the decomposition of integral in Eq. (6.2) into cluster-independent factors. Convergence analysis of $\mathbb{E}\{\Gamma\}$ can be easily performed statistically by the well-known method of the Z-test, since node positions realizations are independent for each Γ sample.



(a) $\sigma = 40$ m (b) $\sigma = 200$ m (c) $\sigma = 600$ m

(d) $\sigma = 1200$ m (e) $\sigma = 2400$ m

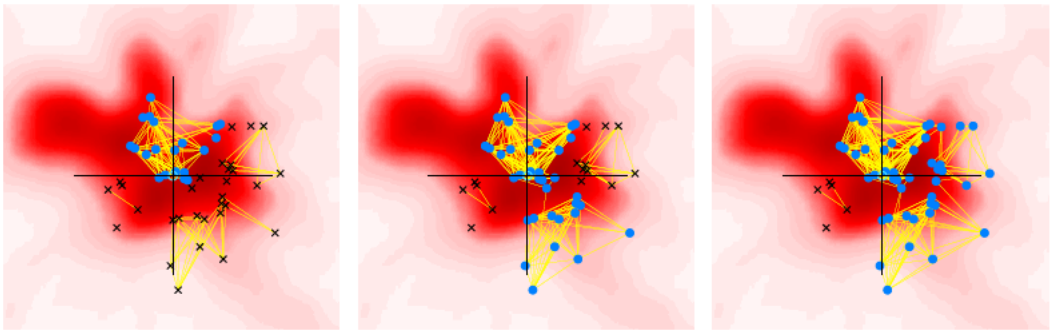
Figure 6.3: Effect of sensor dispersion in the importance covered varying σ. Importance is only captured in circled areas

6.4.4. Sensing-connectivity balance examples

To finish this section we present two examples to clarify how the different parameters affect the network coverage in a single cluster deployment. On the one hand, Figure 6.3 shows deployments for different σ values and fixed $P_{\text{tx}} = -20$ dBm in the surface elevation map of Bullialdus crater of Figure 6.1. Active nodes are depicted with their sensing areas (blue circles) and isolated nodes are marked with 'x' symbols. Clearly, for low σ values most nodes are connected but sensing areas overlap, and thus deployment is inefficient. For high values of σ the deployment is also inefficient because of the sub-connected topology. On the other hand, Figure 6.4 shows how radio P_{tx} parameter also affects the results, in this case keeping the same node positions.

6.5. Mass model

Payload limitation is one of the top constraints in the design of space exploration missions. It depends on the rocket lift capacity and on the mission requirements (e.g. orbit type). As example of its variability, Table 6.2 summarizes payload and characteristics of different Moon's missions up to date.



(a) $P_{\text{tx}} = -20$ dBm (b) $P_{\text{tx}} = 0$ dBm (c) $P_{\text{tx}} = 20$ dBm

Figure 6.4: Effect of sensor dispersion in the importance covered at fixed σ and varying P_{tx}

Mission	Year	Mission Type	Mass Description	
LRO	2009	Lunar polar orbiter	Satellite dry mass	949 Kg
LCROSS	2009	Lunar orbiter	Satellite dry mass	534 Kg
			Propellant	300 Kg
Chandrayaan-1	2008	Lunar orbiter	Spacecraft dry mass	523 Kg
			- Scientific payload	55 Kg
			- Moon Impact Probe	35 Kg
			Propellant	70 Kg
Clementine	1994	Lunar orbiter	Spacecraft dry mass	227 Kg
			Propellant	197 Kg
Lunokhod 2	1973	Lunar orbiter & lander	Rover	840 Kg
			Lander	974 Kg
Apollo XI	1969	Lunar orbiter & lander	Command Module	5,557 Kg
			Service Module	23,244 Kg
			Total Lunar Module	15,065 Kg
			- Dry mass Ascent Stage	2,180 Kg
			- Initial propellant onboard	2,639 Kg
			- Dry mass Descent Stage	2,034 Kg
			- Initial propellant onboard	8,212 Kg

Table 6.2: Payload description of Lunar missions. Data extracted from NASA National Space Science Data Center (NSSDC)

Therefore, in order to define the mission, a payload limit P must be assumed, representing the total mass that a launcher can deliver to a predefined insertion point $(\boldsymbol{x}_0, h_0)$ at an insertion velocity v_0. While v_0 is assumed fixed, a slight variation in $(\boldsymbol{x}_0, h_0)$ is considered possible. Thus, we assume $(\boldsymbol{x}_0, h_0)$ may be freely selected in $\mathcal{X} \times \mathbb{R}_0^+$. Figure 6.2 depicts these parameters and variables in the insertion point of the deployment sequence.

In our case, the payload must include the total mass of the nodes (scientific load), m^l, the structural mass of the deployment probes (structural mass), m^s, and the mass of the propellants required by the deployment probes to reach their designated deployment points (propellant mass), m^p. Let us denote $m_i = m_i^l + m_i^s + m_i^p$.

Hence, mass constraint is given by:

$$\sum_{i=0}^{c} m_i \leq P \tag{6.8}$$

Scientific payload can be related straightforwardly to the number of nodes in each cluster (n_i) and the mass of a single node (m_{node}) as $m_i^l = m_{\text{node}} n_i$. Each node is composed by several main parts: physical sensor, electronics (processor, circuitry, antenna, etc.), battery, heating system, and isolator and protective layers. Total mass can be decomposed in a constant hardware part, m_{hardware}, which includes all of the previous systems, and the batteries mass, m_{battery}, which is variable and depends on the target mission lifetime T (see Section 6.6). Therefore, $m_{\text{node}} = m_{\text{hardware}} + m_{\text{battery}}$ (see Table 6.1 for mass examples).

Besides, the structural mass will depend on the scientific load to be transported as well as on the propellant required for doing so: a higher number of nodes or more propellant will demand more volume and consequently more mass. In rocket designs, two ratios relating masses are commonly defined, the Structural Coefficient, $\epsilon_i = m_i^s/(m_i^s + m_i^p)$, and the Payload Ratio, $\lambda_i = m_l/(m_s + m_p)$. In our design we have assumed a constant, cluster-independent, $\gamma = \epsilon_i/\lambda_i$ figure. This way, the structural mass can be computed as $m_i^s = \gamma m_i^l$.

Computation of the propellant mass require insight into the Tsiolkovsky rocket equation (6.9), which describes the motion of vehicles that follow the basic principle of a rocket: a device that can apply acceleration to itself (a thrust) by expelling part of its mass with high speed and move due to the conservation of momentum. Tsiolkovsky equation is given by:

$$\|\boldsymbol{\Delta v}\| = v_e \ln \frac{m_0}{m_1} \tag{6.9}$$

Where,

- $\|\boldsymbol{\Delta v}\|$ is the maximum change of speed of the rocket if no other external forces act, computed as the sum of all $\boldsymbol{\Delta v}$ norm-2 contributions.
- m_0 is the initial mass before the expulsion of propellant.
- m_1 is the final mass after the expulsion of propellant.
- v_e is the average exhaust velocity, a rocket parameter.

v_e is related to the specific impulse I_{sp}, which is an indicator of engine efficiency defined as the change in momentum per unit mass of rocket propellant, and to the gravitational acceleration constant g_0 (see Table 6.1).

$$v_e = |g_0| I_{sp} \tag{6.10}$$

Next we compute the $\boldsymbol{\Delta v}$ budget associated to the mission profile proposed, in order to compute the total propellant required.

6.5.1. $\boldsymbol{\Delta v}$ budget computation

In our context, each deployment probe is a rocket device that shall complete several maneuvers during the deployment, as indicated in Figure 6.2. Performing these maneuvers require modifications in the trajectory by using the rocket engine. We have considered an *impulsive maneuvers* model where instantaneous changes in the spacecraft's velocity are assumed possible. Although impulsive maneuvers are not feasible in a real probe, they are regarded as a good approximations to the real trajectory, reducing greatly the complexity of calculations. The maneuvers considered are:

1. From the insertion point $(\boldsymbol{x}_0, h_0)$, departing with an insertion velocity $\boldsymbol{v}_0$ at time t_0 with a total mass m_i, the ith deployment probe applies a $\boldsymbol{\Delta v}_{i,0}$ that modifies its trajectory to reach the corresponding deployment point $(\boldsymbol{x}_i, h_i)$ at time $t_{i,1}$. That is,

$$(\boldsymbol{x}_i, h_i) = (\boldsymbol{x}_0, h_0) + t_{i,1}(\boldsymbol{v}_0 + \boldsymbol{\Delta v}_{i,0}) + \frac{1}{2}\boldsymbol{g}t_{i,1}^2 \tag{6.11}$$

   where $\boldsymbol{g}$ is vector $(0, 0, g_0)$. Therefore $\boldsymbol{\Delta v}_{i,0}$ can be expressed as:

$$\boldsymbol{\Delta v}_{i,0} = \frac{(\boldsymbol{x}_i, h_i) - (\boldsymbol{x}_0, h_0)}{t_{i,1}} - \frac{1}{2}\boldsymbol{g}t_{i,1} - \boldsymbol{v}_0 \tag{6.12}$$

2. Deployment point is reached with velocity $\boldsymbol{v}_{i,1} = \boldsymbol{v}_0 + \boldsymbol{\Delta v}_0 + \boldsymbol{g}t_{i,1}$, and must be adjusted to the release velocity, $\boldsymbol{v}_r = 0$, according to the release mechanism configuration described in Sect. 6.4.1. Therefore a $\boldsymbol{\Delta v}_{i,1}{=}{-}\boldsymbol{v}_{i,1}$ shall be applied at time $t_{i,1}$:

$$\boldsymbol{\Delta v}_{i,1} = -\boldsymbol{v}_0 - \boldsymbol{\Delta v}_0 - \boldsymbol{g}t_{i,1} \tag{6.13}$$

   Let us denote total mass before nodes expulsion as $m_{i,1}$. After expulsion the probe mass is decreased by the scientific payload, hence $m_{i,1^+} = m_{i,1} - m_i^l$

3. After nodes expulsion, the probe without load must land at surface, height z_i, from the release height, h_i. The probe is assumed to fall freely from this release altitude (i.e. with vector acceleration $\boldsymbol{g}$), and a final impulsive maneuver is required to stop probe just before hitting ground to land softly. The velocity just before landing is denoted as $\boldsymbol{v}_{i,2}$. Thus,

$$\boldsymbol{\Delta v}_{i,2} = -\boldsymbol{v}_{i,2} = -(0, 0, \sqrt{2g_0(z_i - h_i)}) \tag{6.14}$$

The probe depletes all propellant with this maneuver, hence the final mass after landing matches the structural mass, i.e. $m_{i,2} = m_i^s$.

Different $\boldsymbol{\Delta v}_{i,0}, \boldsymbol{\Delta v}_{i,1}, \boldsymbol{\Delta v}_{i,2}$ maneuvers are compatible with the profile given above. Among them, the one with minimal propellant requirement must be selected due to the mission payload limitation. Since the consumption for the last maneuver is constant, we aim at searching the time $t_{i,1} = t^*$ fulfilling:

$$t^* = \arg\min_{t_{i,1}} \{\|\boldsymbol{\Delta v}_{i,0}\| + \|\boldsymbol{\Delta v}_{i,1}\|\}, \text{ s.t. } t_{i,1} \in \mathbb{R}_0^+ \tag{6.15}$$

This optimum can be easily computed in practice by expressing analytically the derivative of the previous function, equaling it to 0 and solving numerically the associated system. The expression obtained for the derivative is very long, thus we are not providing it explicitly, but it can be obtained straightforwardly with the aid of symbolic calculus tools like Mathematica.

The total mass m_i of the i*th* probe can be related to mass $m_{i,1}$ by means of Tsiolkovsky-equation:

$$m_i = m_{i,1} e^{\frac{\|\boldsymbol{\Delta v}_{i,0}\| + \|\boldsymbol{\Delta v}_{i,1}\|}{v_e}} \tag{6.16}$$

where $\boldsymbol{\Delta v}_{i,0}, \boldsymbol{\Delta v}_{i,1}$ are computed using eqs. (6.12) and (6.13) for the optimal time $t_{i,1}$ obtained in eq. (6.15). Similarly, the final mass, m_i^s, is related to $m_{i,1}$ as

$$m_{i,1} = m_i^l + m_i^s e^{\frac{\|\boldsymbol{\Delta v}_{i,2}\|}{v_e}} \tag{6.17}$$

And since $m_i^l = m_{\text{node}} n_i$ and $m_i^s = \gamma m_i^l$, the total mass constraint can be expressed as

$$m_{\text{node}} \sum_{i=0}^{c} n_i (1 + \gamma e^{\frac{\|\boldsymbol{\Delta v}_{i,2}\|}{v_e}}) e^{\frac{\|\boldsymbol{\Delta v}_{i,0}\| + \|\boldsymbol{\Delta v}_{i,1}\|}{v_e}} \leq P \tag{6.18}$$

The last inequality represents the *mass constraint* of our problem. It depends exclusively on the variables of our optimization problem, i.e. number of nodes per cluster, n_i, their deployment position (x_i, h_i), and the insertion position (x_0, h_0). In addition, a number of parameters, summarized in Table 6.1, allow to customize the deployment behavior.

Finally, let us remark that different maneuvering models could have been also considered by computing their corresponding $\boldsymbol{\Delta v}$ budget and applying the methodology provided in this section to compute a similar mass constraint.

6.6. Energy model

As pointed out, mission has to last a minimum time T with very different ambient conditions which will impact design. As a reference, if we assume a mission length covering a whole Moon's synodic month (on average 29.53 Earth days), surface temperature can range from 40 K (night) to 396 K (day). Nodes are considered to be battery powered. Other options like solar arrays are discarded due to the power unavailability during lunar night. Existing ultra-low-high-temperature batteries or other upcoming battery technologies must be used to meet the mass and size requirements. For example, military-design lithium thionyl chloride batteries have a high energy density ($E = 500$ Wh/kg) and allow a high current drain rate cell with extended temperature range (from 218 K to 398 K) [Pilarzyk, 1997]. In order to keep batteries temperature above the minimum threshold of 218 K, heater-modules and thermal isolators will be required.

Batteries capacity will depend on the required operation time T and on the current drain, which in turn depends on the duty cycle of the WSN. Let us denote I_{active} and I_{sleep} the current drain during each phase. The heating system has a current drain of I_{heater} during activity and a particular ratio of activity, ρ_{heater}. Thus, there are D=4 different levels of current drain and activity ratio:

1. Node active, heater off: $I_1 = I_{\text{active}}$, $\rho_1 = \rho(1 - \rho_{\text{heater}})$.
2. Node active, heater active: $I_2 = I_{\text{active}} + I_{\text{heater}}$, $\rho_2 = \rho\rho_{\text{heater}}$.
3. Node sleep, heater active: $I_3 = I_{\text{sleep}} + I_{\text{heater}}$, $\rho_3 = (1 - \rho)\rho_{\text{heater}}$.
4. Node sleep, heater off: $I_4 = I_{\text{sleep}}$, $\rho_4 = (1 - \rho)(1 - \rho_{\text{heater}})$.

Reference values for the node and heater consumptions are given in Table 6.1.

A common approximation to compute the battery capacity is given by Peukert's law[1],

$$C = T \sum_{k=1}^{D} I_k^{\zeta} \rho_i \tag{6.19}$$

where C is the capacity at a one-ampere discharge rate and ζ is the Peukert's exponent, which indicates how well a battery performs under continuous heavy discharge current.

By applying the previous equation the minimum required capacity C can be obtained for a given T constraint. This is related to our deployment model since the battery mass, which depends on C, must be added to the mass budget of a node

[1]More advanced models, e.g. based on ragone plots can be used as well, if they are available for the battery model.

	0 dBm		-20 dBm	
	m_{battery}	C	m_{battery}	C
$\rho = 0.01$	27.59 g	3,780 mAh	27.24 g	3,732 mAh
$\rho = 0.1$	43.32 g	5,934 mAh	39.83 g	5,457 mAh

Table 6.3: Estimated battery mass and capacity. $E = 500$ Wh/kg, $T = 708$ h, $\zeta = 1$.

(see Section 6.5): $m_{\text{battery}} = \frac{CV}{E}$, where V is the open circuit voltage of the node and E is the energy density of the battery technology. Table 6.3 provides examples for the capacity and battery mass as a function of ρ. In addition, the heating system and thermal isolator mass is considered part of the m_{hardware} constant term of the nodes hardware, as described in Section 6.5.

6.7. Formal optimization problem statement

Given,

$$\alpha : \mathcal{X} \to \mathbb{R}_0^+$$
$$z : \mathcal{X} \to \mathbb{R}_0^+$$

Find variables:

$$c$$
$$n_i, x_i, h_i, \text{ for } i = 0, \ldots, c$$

such that:

$$\mathbb{E}\{\int_{\mathcal{B}} \alpha(x)\mathrm{d}x\} \tag{6.20}$$

is maximized, and subject to:

1. $c \in \mathbb{N}^+$
2. $n_i \in \mathbb{N}^+ \ \ \forall i = 1, \ldots, c$
3. $x_i \in \mathcal{X} \subset \mathbb{R}^2 \ \ \forall i = 1, \ldots, c$
4. $h_i \in \mathbb{R}_0^+ \ \ \forall i = 1, \ldots, c$
5. $m_{\text{node}} \sum_{i=0}^{c} n_i (1 + \gamma e^{\frac{\|\boldsymbol{\Delta v}_{i,2}\|}{v_e}}) e^{\frac{\|\boldsymbol{\Delta v}_{i,0}\| + \|\boldsymbol{\Delta v}_{i,1}\|}{v_e}} \leq P$

6.8. Optimization problem solving

The previous is a Mixed-Integer Non-linear Non-Convex mathematical problem (NP-hard), whose solution can only be approached by using stochastic solvers. In particular our solver employs the Simulated Annealing (SA) meta-heuristic. SA continuously tries random variations of the current solution and accepts new ones either if they improve the current solution or randomly (with a probability that smoothly rises down as the algorithm progresses) if not. This way of dealing with solution renewal allows the SA to escape from local minima. Although there is no guarantee that the final solution corresponds to a global optima, SA is able to find coherent and acceptable solutions to many problems given that a proper algorithm to generate new solutions from the previous ones is found. This question is critical in our case since evaluating a vast amount of solutions would be computationally prohibitive by using the method based on Monte Carlo evaluation provided in Section 6.4.3. The main problem with this method is that it requires reevaluating all the network connectivity and sensing per Γ realization, which seems inefficient when only local changes are applied to a given cluster. A way of simplifying the structure of the Monte Carlo evaluation, and proposing a suitable SA solution generation algorithm, stems from the fact that the optimization goal in Eq. (6.20) is equivalent to:

$$\mathbb{E}\{\Gamma\} = \int_{\mathcal{X}} \alpha(x) Pr[x \in \mathcal{B}] \mathrm{d}x \tag{6.21}$$

where $Pr[x \in \mathcal{B}]$ can be rewritten as:

$$Pr[x \in \mathcal{B}] = 1 - \prod_{i=1}^{c} (1 - Pr[x \in \mathcal{B}_i]) \tag{6.22}$$

being $\mathcal{B}^i$ the area covered by the cluster i, i.e. $\mathcal{B}_i = \bigcup_{j \in \Phi_i} B(x_i^j, r_s)$.

This expression indicates that if the launch parameters are changing only for a single cluster, then the network coverage can be recomputed *without the need to recalculate the coverages associated to all clusters*. To do so, let us consider a new solution where changes only affect cluster i. Then $Pr[x \in \mathcal{B}]$ can be recomputed in two steps:

1. Old cluster removing:

$$Pr[x \in \mathcal{B}^*] = \frac{Pr[x \in \mathcal{B}] - Pr[x \in \mathcal{B}_{i,\mathrm{old}}]}{1 - Pr[x \in \mathcal{B}_{i,\mathrm{old}}]} \tag{6.23}$$

2. New cluster adding:

$$Pr[x \in \mathcal{B}] = 1 - (1 - Pr[x \in \mathcal{B}^*])(1 - Pr[x \in \mathcal{B}_{i,\text{new}}]) \tag{6.24}$$

Computation of $Pr[x \in \mathcal{B}_{i,\text{new}}]$ must be performed by means of Monte Carlo evaluations, following similar steps to the general network-updating procedure of Section 6.4.3, but for a single cluster and substituting steps from 4) to 7) by:

4. Determine the set $\mathcal{B}_i$ given by Eq. (6.3).
5. Set $p(x)$ realization as 1 if $x \in \mathcal{B}_{i,\text{new}}$ or 0 otherwise, for each $x \in \mathcal{X}$
6. Repeat from 3) until $\overline{p(x)}$ converges for each $x \in \mathcal{X}$
7. Return $Pr[x \in \mathcal{B}_{i,\text{new}}] \leftarrow \overline{p(x)}$

Then $\mathbb{E}\{\Gamma\}$ is evaluated through expression Eq. (6.21). This procedure allows a dramatic reduction in the computational demands of the SA, leading to a feasible solver.

Let us note that the two final steps in the previous procedure would require determining a non-numerable set Φ_i, as well as determining set membership over *all* points the non-numerable set $\mathcal{X}$. In practice, a finite regular Cartesian grid $\boldsymbol{X} \subset \mathcal{X}$ is assumed both for input data and for internal computations. That is, input mappings α and z are provided as matrices. Values for the points outside the grid are determined internally by linear interpolation. Similarly, sets $\mathcal{B}_i$ are internally stored as matrices, whose cells have either value 1 or 0. The mapping $p(x)$ is also determined for points in the grid $\boldsymbol{X}$, hence it can be also stored as a matrix, which convergence is checked for each point by means of Z-test, assuming it is achieved when the confidence interval of $\pm 1\%$ of the numerical average has a confidence level at least of 99%. Test have shown that a small number of iterations are required (in the order of 100).

6.8.1. Solution generation algorithm

Table 6.4 describes the updating rules for generation of new solutions from the current one $S = \{x_0, h_0, c\} \cup \{x_i, h_i\}_{i=1,\ldots,c}$. Note that since h_i and σ_i are univocally related through Eq. $\sigma_i = \beta\delta_i$ with δ_i given in Eq. (6.4) rules of Table 6.4 straightforwardly operates with σ_i instead.

Finally, a note must be added in relation to the goal function used by the SA implementation. Although the problem expression provided in Section 6.7 can be solved, we have observed that a variation of the problem relaxing constraint 5) by adding it to the goal function leads to a much faster convergence. The proposed goal function to maximize is:

$$\int_{\chi} \alpha'(x) p(x \in \mathcal{B})\, \mathrm{d}x$$

where

$$\alpha'(x) = \begin{cases} \alpha(x) - \varepsilon(\Delta P) & \Delta P \leq 0 \\ \alpha(x) - \Upsilon(1 + \Delta P) & \Delta P > 0 \end{cases} \tag{6.25}$$

where ΔP is the mass excess of the solution, i.e. the left hand-side term in Eq. (6.18) minus the mass limit P, Υ is a big factor that penalizes the mass excess of a solution (the term 1 is present to guarantee a minimal penalty even with small ΔP). With this goal function, the SA may temporarily accept solutions with a mass excess (escaping from local minima solutions), which afterwards is eliminated since it largely penalizes the goal function. Furthermore, ε is a factor several orders of magnitude less than $\alpha(x)$ that forces having feasible solutions with ΔP close to 0. This way solutions tend to have more mass, leading to more nodes and consequently more sensed surface. In all the tests performed using this alternative goal function, the solution convergence time had a 100-fold reduction, compared to the original goal function.

Regarding the computational requirements, we have done our tests in an AMD Opteron@2600 MHz 48 multi-core system with 64GB of memory. In this machine the average computing time for each iteration was 4.36 s, though large variations were possible depending on the number of clusters and nodes per cluster. Besides, the average total time until SA convergence was in the 90% of the cases in the range 46.65 ± 7.45 hours, being the average number of iterations in the order of 50000 though also with a high variability. But we shall note that several repetitions of the SA were necessary (in the order of 30), like in any optimization complex problem, due to the random nature of the SA.

Reference values for the solution updating rules parameters are provided in Table 6.1. The internal parameters of temperature and maximal iterations for the SA are also provided in Table 6.1.

6.9. Optimal deployments in the Bullialdus crater

We have applied the optimization methodology developed in the previous sections to the scenario of the Bullialdus crater described in Section 6.1. In this case $\boldsymbol{X}$ represents a 630×600 pixels map, which corresponds to a 40 meters per pixel scale, totalling a real scenario size of 25.2 x 24 Km. The network lifetime T has been established to the average synodic month (29.53 Earth days), and the sensing radio r_s has been set to 80 m (2 px.).

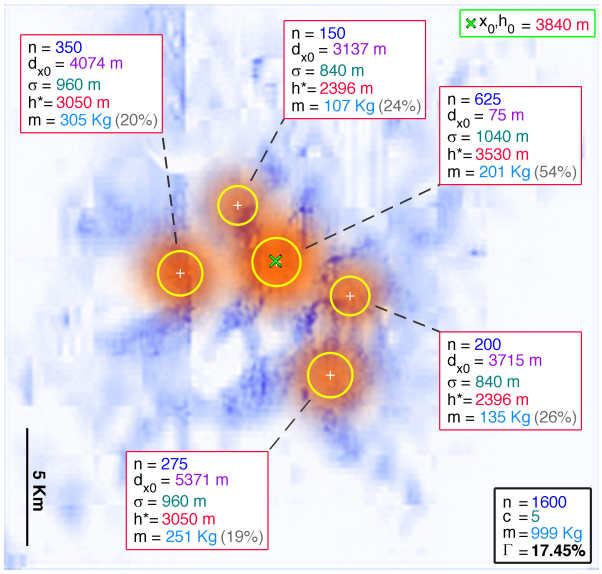
Figures 6.5 through 6.8 show the optimization results for variations of the two main WSN communication parameters, the duty cycle (ρ) and the transmission

%	**Rule action**
20%	$i \leftarrow U[1, c_{\max}]$ $n_i \leftarrow \Delta n \cdot U[0, \lfloor \frac{n_{\max}}{\Delta n} \rfloor]$ with $\sum_{k=1}^{c_{\max}} n_k \neq 0$ $\sigma_i \leftarrow U(\sigma_{\min}, \sigma_{\max})$
10%	$i \leftarrow U[1, c_{\max}]$ $n_i \leftarrow \Delta n \cdot U[0, \lfloor \frac{n_{\max}}{\Delta n} \rfloor]$ with $\sum_{k=1}^{c_{\max}} n_k \neq 0$
10%	$i \leftarrow U[1, c_{\max}]$ $\sigma_i \leftarrow U(\sigma_{\min}, \sigma_{\max})$
35%	$i \leftarrow U[1, c_{\max}]$ $x_i \leftarrow x_i + N(0, \frac{\sigma_i}{2} I)$
3%	$i \leftarrow U[1, c_{\max}]$ $x_i \leftarrow$ uniformly random position within $\mathcal{X}$
20%	$x_0 \leftarrow x_0 + N(0, \sigma_0 I)$ $h_0 \leftarrow h_0 + N(0, \sigma_0)$
2%	$x_0 \leftarrow$ uniformly random position within $\mathcal{X}$ $h_0 \leftarrow h_0 + N(0, \sigma_0)$

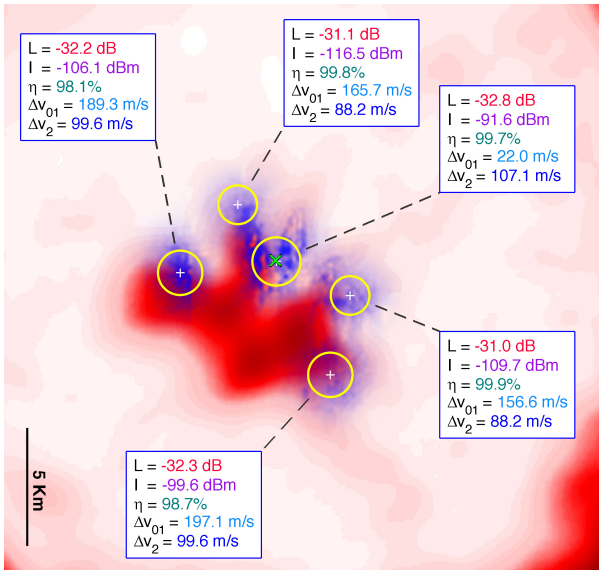
Table 6.4: Rules for generation of a new solution. First column indicates the percentage of times that the rule is applied. Algorithm parameters are indicated in Table 6.1.

power (P_{tx}). In these experiments the mass limit P was in all cases of 1000 Kg. In each figure, the leftmost (blue) map shows the importance map, the launch positions and the dispersion associated to each cluster. A box indicates for each cluster the release height and a summary of the main cluster indicators. Moreover, superimposed to the importance it is shown the probability map (in orange) $Pr[x \in \mathcal{B}]$, which provides a clear indication of the network coverage of the optimal solution. The map in the middle shows the elevation map, and parameters about the radio propagation in each cluster, the average communication path-loss (L), the average interference level (I), the average number of nodes connected to the sink in the cluster (η), and the Δv modulus of the trajectory corrections. This figure also shows (in blue) the map $\alpha(x)Pr[x \in \mathcal{B}]$, which represents the amount of information actually captured by the network. Finally, the rightmost figure shows a 3D view of the solution. The green mark represents the insertion point, whereas the yellow ones are the deployment positions for each cluster.

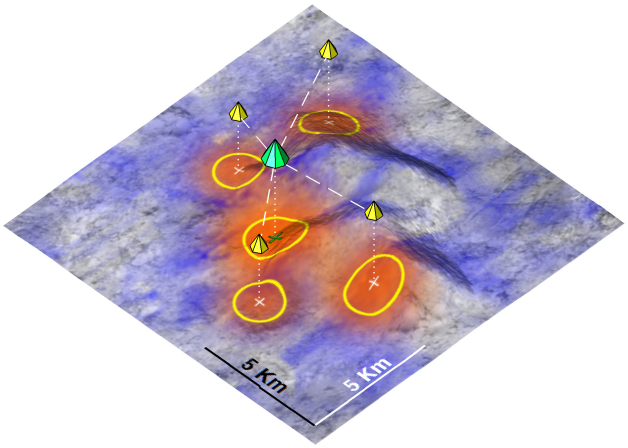
Solution complexity clearly rules out the possibility to find them by means of educated guesses. Moreover, changes in any of the communication parameters produces totally different solutions. Some effects are clearly observed. First, the insertion point is located close to the barycenter of the networks, which seems rational for a propellant minimization strategy. Second, the higher the duty cycle is, the higher is the number of clusters. This seems caused because the increase in the duty cycle leads to a higher level of interferences. A good alternative to reduce interferences is operating in separate clusters which, as previously said, use different



(a) Importance map

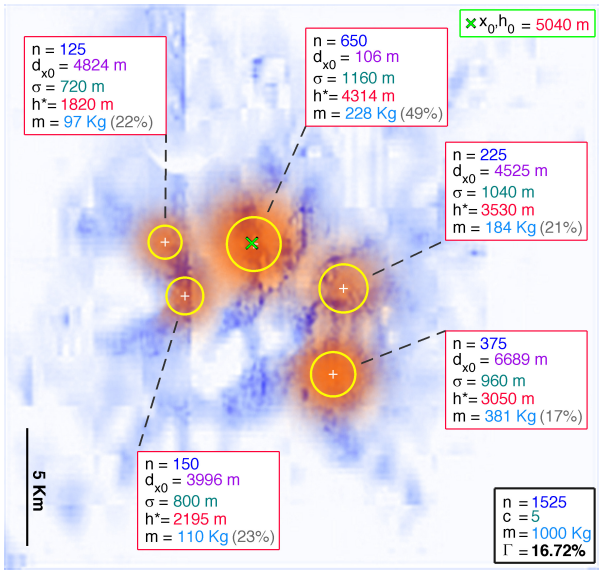


(b) Elevation map

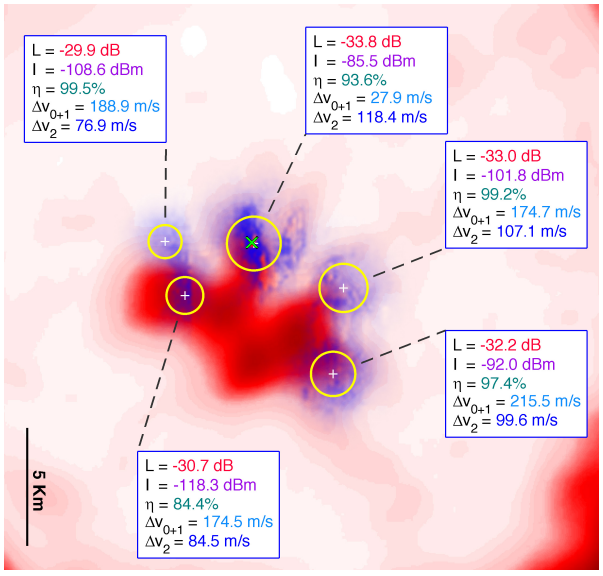


(c) 3D map

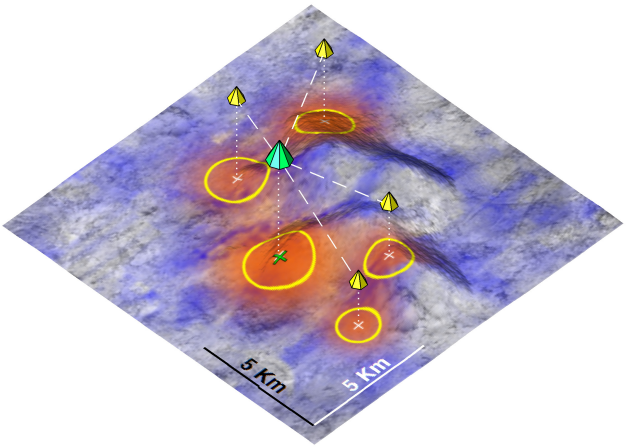
Figure 6.5: Optimal solution, $P_{\text{tx}} = 0$ dBm, $\rho = 0.01$



(a) Importance map

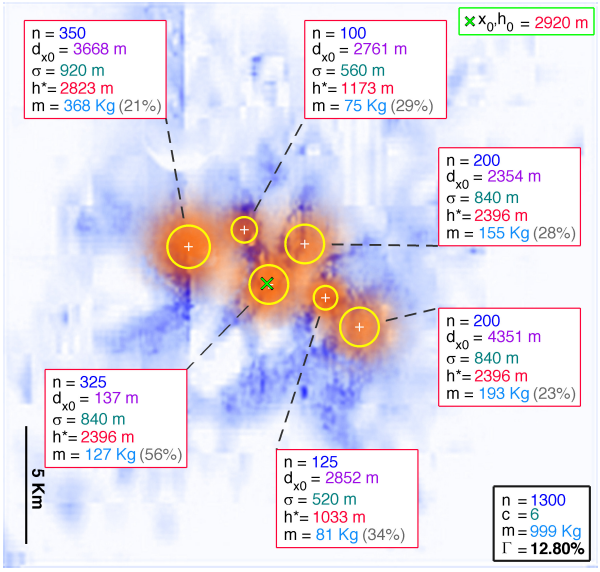


(b) Elevation map

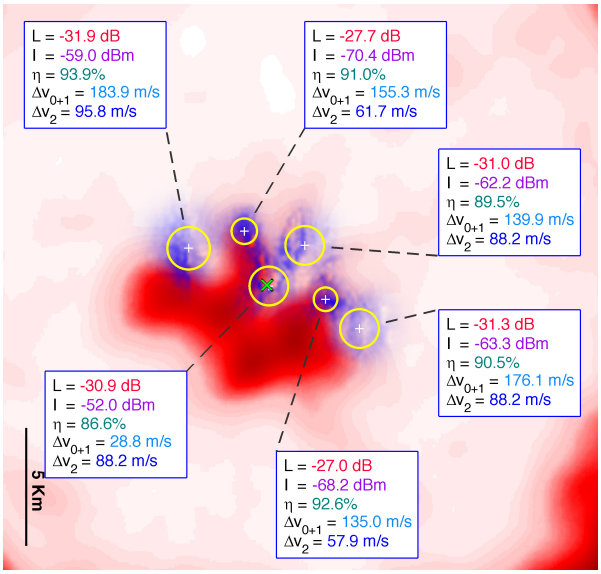


(c) 3D map

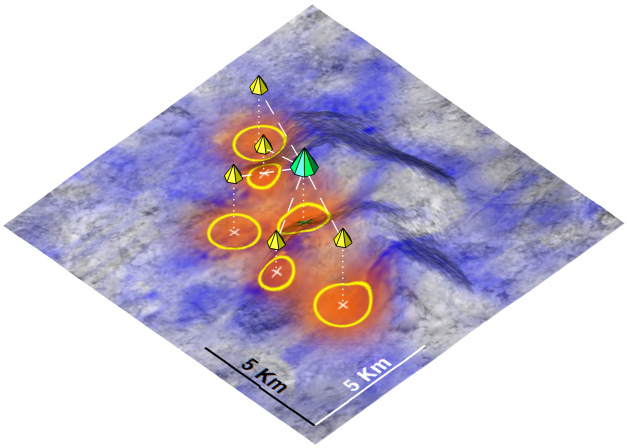
Figure 6.6: Optimal solution, $P_{\text{tx}} = -20$ dBm, $\rho = 0.01$



(a) Importance map

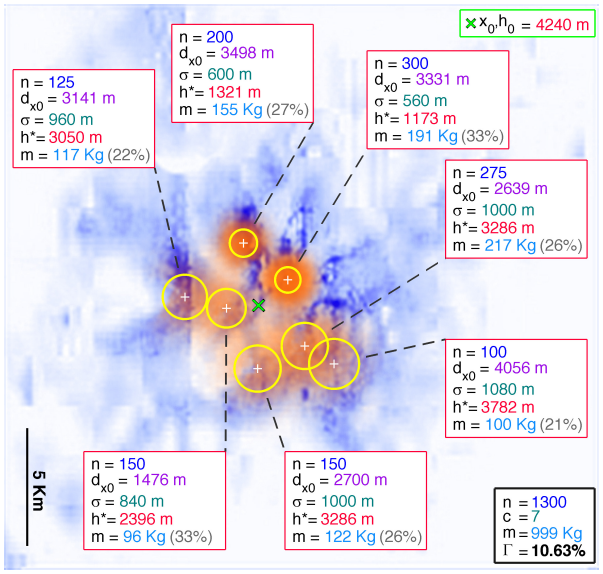


(b) Elevation map

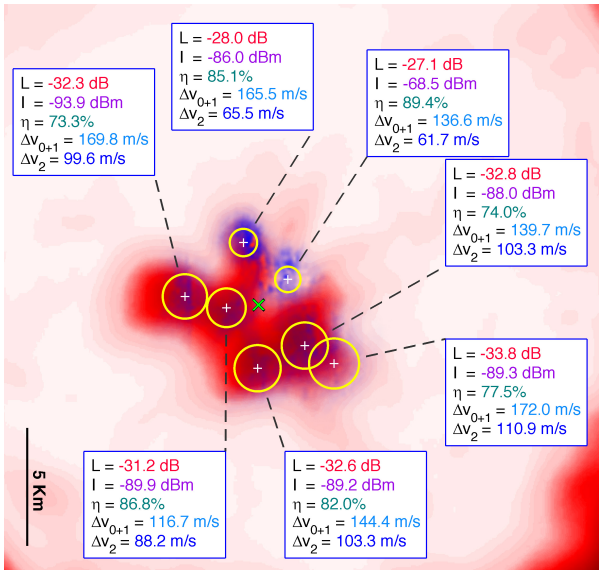


(c) 3D map

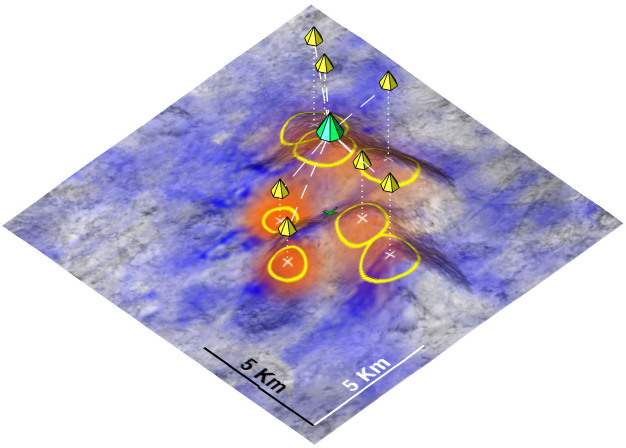
Figure 6.7: Optimal solution, $P_{\text{tx}} = 0$ dBm, $\rho = 0.1$



(a) Importance map

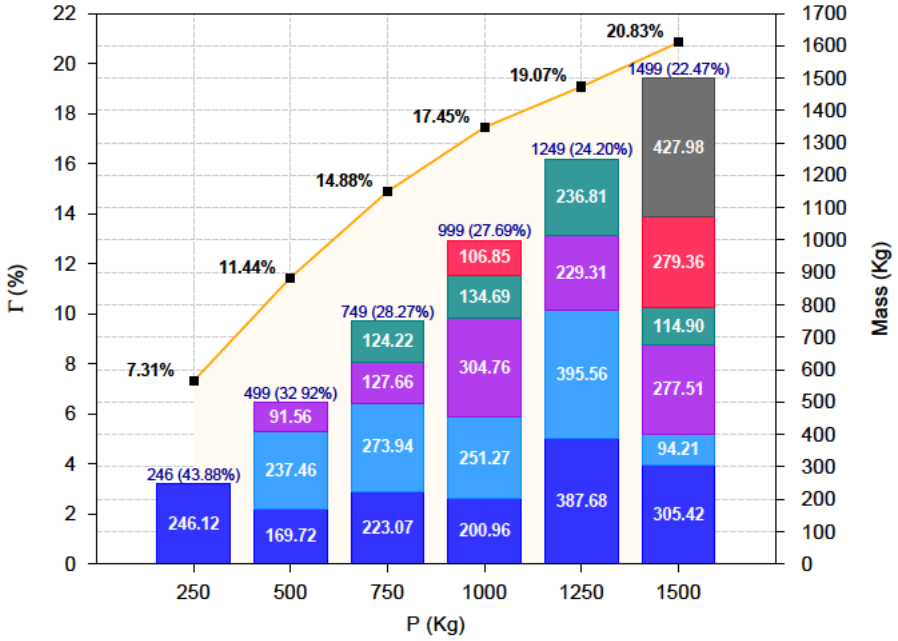


(b) Elevation map

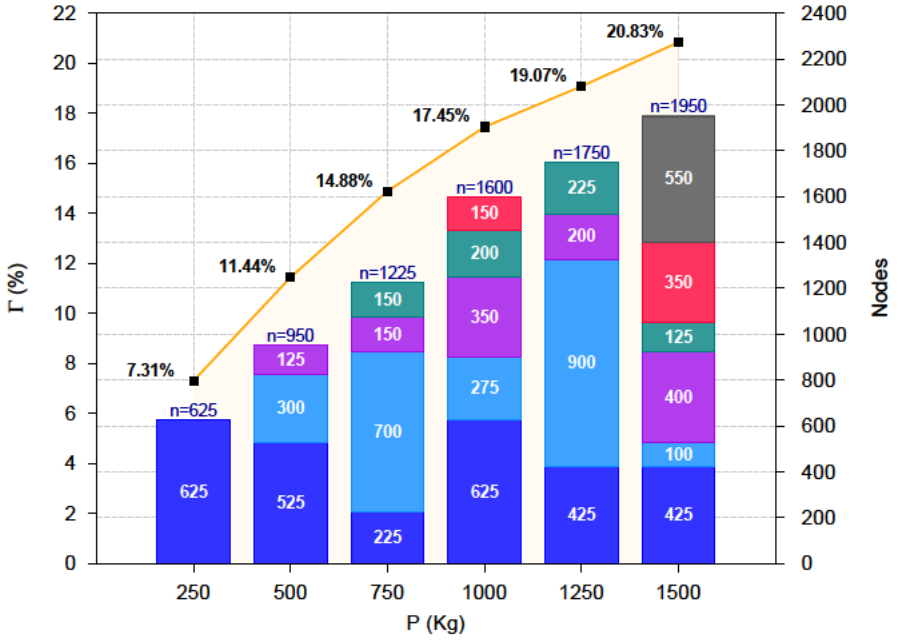


(c) 3D map

Figure 6.8: Optimal solution, $P_{\text{tx}} = -20$ dBm, $\rho = 0.1$



(a) Mass



(b) Nodes

Figure 6.9: Coverage versus payload limit P

frequency channels and do not interfere each other. Reduction of the transmission power also decrements the sensing coverage, e.g. from 17.45% in the P_{tx} = -20 dBm, $\rho = 0.01$ to 16.72% when output power decreases to 0 dBm. Identically, for the $\rho = 0.1$ configuration, the coverage reduces from 12.80% to 10.63%. A direct cause for it is the reduction in the received power. To a lesser extent, the interference levels are reduced as well, although the first effect is dominant. This is more noticeable with high duty cycle ratios, as in Figs. 6.7 and 6.8.

Figure 6.9 shows the coverage versus the payload limit P. Non-surprisingly it increments if the payload can be higher, although in a sub-linear regime. This is because it is increasingly difficult to gain coverage, since best sites tends to be covered first, reducing the opportunities of selecting good sites afterwards. In addition, Figure 6.9 also shows the mass and node distribution per cluster, also versus P. On top of each bar in the mass figure the percentage of non-propellant mass is indicated. It decreases due to the same effect described before, the need to reach more difficult launching sites also demands extra propellant, thus reducing the mass ratio.

Finally, Figure 6.10 shows a Quality Function Deployment (QFD) diagram with the main interrelationships among the deployment parameters and properties. For each row, the QFD describes the effect that an increment produces in the property referred in the column. For example, when the transmit power is increased, the resulting optimal deployment tends to have a similar or lesser number of clusters. Another example comes when the duty cycle increases, which results in a worse connectivity due to a high increase in the interference level. In addition the QFD shows in the top the correlation among the deployment properties, e.g. there is a strong correlation between connectivity and interference, or between interference and number of active nodes. An example of weak correlation appears e.g. between network coverage and battery capacity. In summary, the QFD offers an analysis at a glance of the main interactions between the elements.

6.10. Conclusions

In this chapter we have described from an analytical perspective the deployment of WSN in a lunar survey mission, although most of the results can be extrapolated to other solar system bodies. A number of novel aspects have been introduced such as the use of suitable propagation models taking into account the 3D terrain, the heterogeneous relevance on the map, detailed lifetime and mass calculations, and the assumption of a mass-limited scenario.

This analytical model allowed us to develop an optimization problem to find the best deployment parameters. In our case, the landing deployment points and release heights, as well as the best trajectory maneuvers for each deployment probe. A computationally efficient method, based on SA, is also proposed with the aim of

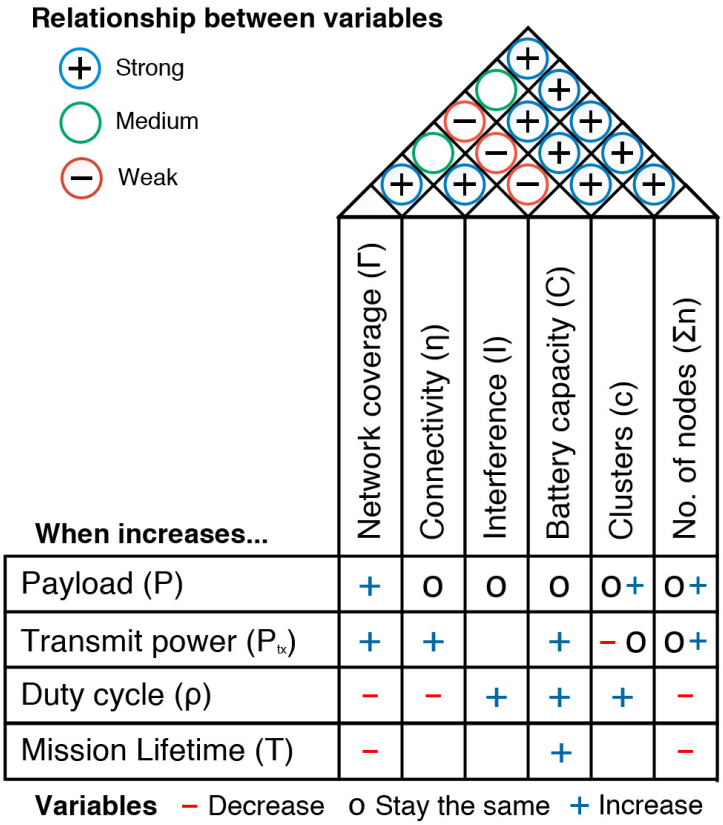


Figure 6.10: Trade-off QFD diagram

solving this problem within a feasible time frame and, although the optimization is highly computing demanding, the solver achieves convergence in the order of only 50 hours, resulting with 30 SA runs in a solving time of approximately 2 months. This must be confronted to a execution time of years that would be necessary without the improvements developed in this work.

As future works we aim at considering also atmospheric effects for deployments in Earth or Mars, and a more detailed node ejection mechanism that might result in either non-Gaussian clusters, or Gaussian clusters with a generalized covariance matrix. Also effects of interoperation between clusters are worth to be considered since overlapping has been observed in some of the solutions.

Part III

AmI system for personalized sport training

– You have seen this before, I can tell. You lied to them.
– I would never lie. I willfully participated in a campaign of misinformation.

Recent developments have led to intelligent environments able to anticipate people actions, react intelligently and support them. These environments constitute a new paradigm in information systems that is known as Ambient Intelligence (AmI) [Ducatel et al., 2001]. AmI systems rely on decision-making, and therefore there are ongoing efforts to apply machine-learning methodologies to them. AmI examples include spaces for education [Stimpson and Cummings, 2014; Xhafa et al., 2011], smart homes [Makonin et al., 2013; Zamora-Izquierdo et al., 2010], health [Acampora et al., 2013; Garg et al., 2012; Barger et al., 2005], sports [Gaeta et al., 2012; Vales-Alonso et al., 2010], leisure [Heinz et al., 2006; Bannach et al., 2007; Sipper et al., 2007], transportation [Riener et al., 2014; Reason and Crepaldi, 2009] and so forth.

Among the activities where these technologies can be applied, sports may be one of the most benefited. In common training systems, the performance of the athletes is only evaluated at the end of the training session, and sensed data are usually incomplete because generally only human biometrics are analyzed. In consequence,

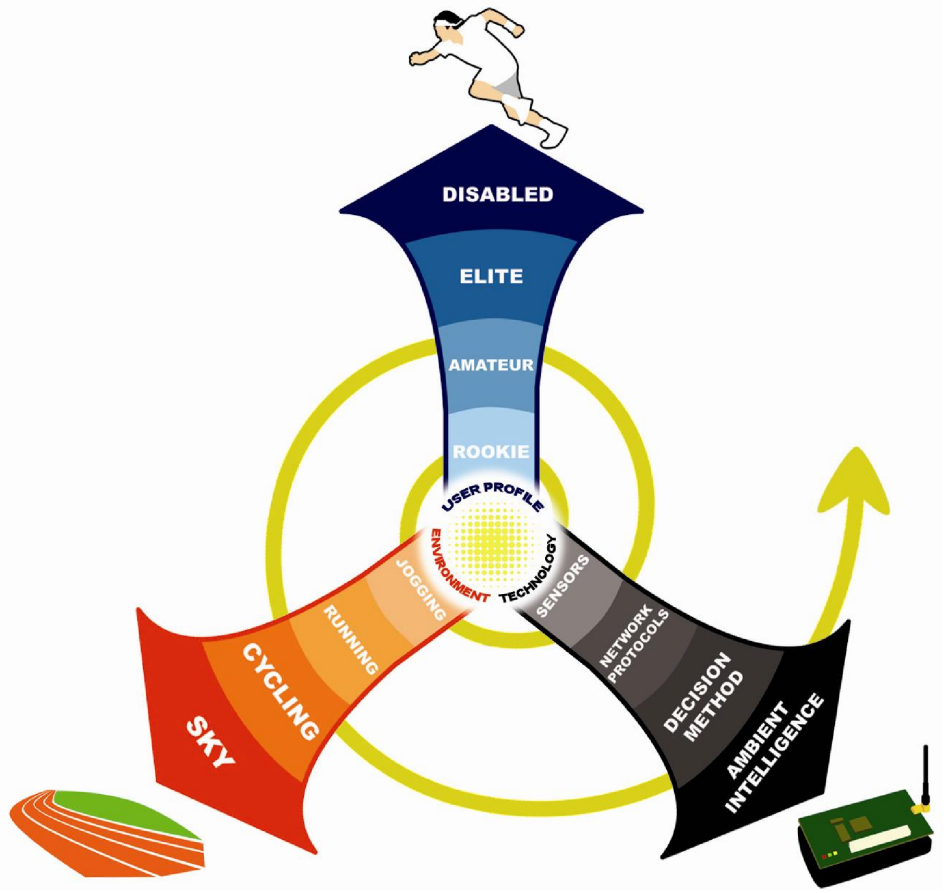


Figure 7.0: General vision of ambient intelligence aimed at Sports

it is not possible to make real-time decisions during the training session, allowing the athlete to accomplish their training objectives.

We propose a system able to provide personalized assistance to athletes in their training. To develop our system, we assume three driving aspects: athletes' profile, environment, and computing and communications/decision technologies, as shown in Figure 7.0. Technology will provide adaptive coordination between the user and the environment, adapting the behavior of the system as it responds to changes in the environment or the user conditions, such as weather or athletes location. Since training conditions can rapidly vary, the system will be expected to make real time decisions to meet the user needs at their full potential. Thus, a decision engine, as well as efficient communications and localization protocols, are two key areas in the development of this training system.

In this context of AmI systems focused on sports, next Chapter introduces SAETA[2], a smart assistant for professional volleyball training based on machine learning techniques. SAETA tackles three main aspects of coaching in elite team sport players:

- Effort control, which controls players' heart rate during customized matches, estimating their temporal evolution by means of different techniques like polynomial regression or Kalman filters.
- Technical-tactical training, which aims at controlling the exercise effort and fatigue levels by means of a Markov Decision Process formulation for the dynamic system. The goal is maximizing the effort while fatigue is avoided.
- Exercise quality training, which complements the former by analyzing the execution of the players' motion. Particular motions are detected by means of a k-Nearest Neighbour classifier and an intensity classification based on previous records.

Globally, SAETA relies on a sensing infrastructure able to monitor both players and environment, which obtains real-time data that is analyzed by different modules on a decision engine. Besides, it has been developed in close collaboration with the *UCAM Volley Murcia* team, which competes in the Spanish premier women's league. Data gathered during training sessions has provided a knowledge base for the algorithms developed, and has been used for the validation of results.

[2]"Sistema Autónomo para el Entrenamiento personalizado de atletas basado en Técnicas de Aprendizaje máquina e inteligencia Ambiental" (*Autonomous system for athletes personalized training based on machine learning and ambient intelligence techniques*), SAETA, 112/UPB 10/11, funded by the Spanish Consejo Superior de Deportes (*National Sports Council*).

Chapter 7

SAETA: A smart coaching assistant for professional volley training

Our cabin attendants are superbly designed... but there's only one problem: when they've served on so many long journeys, fatigue begins to set it. For example, they might want to laugh, but the smile would be slow to come. They might want to cry, but the tear wouldn't well up till the next day...

7.1. Introduction

At this moment wearable computing is a common tool for elite athletics training and it is even reaching the mass market. Wearable computing systems can collect biometrics about pace, heart rate (HR) and blood oxygen concentration, as well as performance related parameters such as body acceleration, distance covered, speed and elapsed training time. However, despite the rich set of available information, these devices tend to be used merely as activity trackers that, at most, give simple recommendations such as "increase pace to cover a predefined distance in time". Nevertheless, elite athletes not only require physical or aerobic supervision, but also guidance to improve exercise intensity and quality (*technical-tactical* training). This task is very complex, even for expert coaches. If we consider a full team with different player roles and variations in training policies during the season, coaching may be daunting.

In short, elite training requires monitoring and evaluating athletes' performance in real-time, and making personalized training decisions, which cannot be tackled

by ordinary training systems. All these challenges have been addressed within the framework of the SAETA project in collaboration with the UCAM Murcia Volleyball team [UCAM, 2014], which plays in the Spanish premier league of women's volleyball. The data obtained during the training sessions has provided a knowledge base for the algorithms and has helped us to validate the proposed control techniques. The main advantages of SAETA include the following:

- Full-time availability.
- Data analysis takes place in real-time, as opposed to current common practices where coaches analyze biometric records at the end of the training session.
- Management of an unlimited number of athletes, complementing the limited vision of a human coach.
- Study of "hidden" variables that may influence athletes' performance. In particular, environmental data, such as humidity.
- Autonomy of operation after the learning phase.

The sensing infrastructure was introduced in a previous work [Vales-Alonso et al., 2010], together with decision engines focused on supporting demonstrative HR control applications in the sport domain. SAETA was developed on top of this infrastructure in the context of volleyball sport. Volleyball was selected mainly because coaches are required to personalize the training of individual players for their different roles in the team. For example, the *libero* or defensive player needs to train muscle groups to achieve a good control of the ball in passing actions so that the team can run the attack. Hitters, on the other hand, must train other muscle groups to improve jumping and ball strike. Therefore, the preparation of a competitive volleyball team requires a high level of control that challenges the work of the coaches.

The approach in SAETA is based mainly on two specialized training studies by [Moras Feliu, 1994] and [Zintl, 1991]. [Moras Feliu, 1994] established six levels for volleyball training, ranging from global body development and strengthening, not specific to volleyball practice (levels 0-2), to actions focused on technical and tactical training (levels 3-5) for volleyball. This is, lower levels address global physical preparation, whereas higher levels are more centered on improving specific player motions. On the other hand, [Zintl, 1991] presented a classification of all training exercises: *continuous*, performing a constant effort over a period; *intervallic*, alternating periods of activity and rest; *repetitional*, repeating a series of similar actions interleaved with complete rest; and *customized match*, playing a match whose intensity is controlled by the coach. From this general perspective, SAETA has been divided in three control modules:

1. Effort control module for customized matches, which controls players' HR by numerically estimating its evolution.

2. Technical-tactical training, which controls the levels of training effort and fatigue by computing the durations of activity and rest periods in repeated exercises.

3. Exercise quality, which controls the quality of the motions by evaluating their intensities.

The last two modules addresses Zintl's *repetitional* exercises and Moras' highest levels of training, respectively. Those modules comprise the most difficult tasks faced by coaches, and are therefore essential for elite team training (see Section 7.2.1). Let us remark that our approach can be extrapolated to other sports (e.g. basketball or soccer), where technical-tactical and aerobic preparation also play an important role [Wrisberg, 2007].

The rest of this Chapter is organized as follows: Section 7.2 briefly reviews the related work in decision-making technologies applied to training guidance and volleyball player preparation; Section 7.3 describes system architecture; Section 7.4 describes the effort control module that estimates the ratios of each effort regime, based on three different methodologies: (i) best-fit polynomial approximations, (ii) Kalman filter and (iii) Sliding-Window distribution estimation; Section 7.5 describes the technical-tactical control system (based on a Dynamic Programming model), which selects the best activity and resting durations in repetitional training, with the goal of maximizing effort while avoiding fatigue; Section 7.6 describes the exercise quality control module that consists of two stages. In the first stage, motions are detected by means of a k-Nearest Neighbors classifier. Then, the intensity of the movement is classified according to recent statistical data from the player. These analysis are reported to the coaches and players in real-time to help them improve training; and finally, Section 7.7 concludes this Chapter by summarizing results and future lines of research

7.2. Related work

7.2.1. Physical performance in sports

There is plenty of work on models for representing the physiological response to training in human performance applied to sports. [Calvert et al., 1976] made early studies about the quantitative relationship between athletic training and performance, stating that there are four features that define human performance in sports, each of them being dominant in different tasks in athletic activities: 1) endurance, 2) strength, 3) skill, and 4) psychological factors. According to the results

of their research, the performance of an athlete is a function of both fatigue and fitness, where fatigue becomes dominant as the performance of the individual approaches the upper limit of his/her physical capacity. Also, their model was able to predict and explain an improvement in performance (for swimmers in their studies) once training stopped, due to the interplay between fitness and fatigue. This is the same foundation of the work by [Moras Feliu, 1994].

Further investigation into the subject was conducted by several authors. For example, [Cooper, 1991] led a series of tests in order to describe a general approach to the modelling of physical performance. In his work, HR data was collected from a wheelchair racer during a whole competition season, including aerobic training, strength training and performance training. HR information was used as a valid estimator of metabolic activity, due to the practical unfeasibility of more precise measurements such as blood lactate level or micro-ventilation rate. The research concluded that human response to training is nonlinear by nature, and recommending adaptive NARMAX [Nelles, 2001] or other nonlinear models as the most useful ones to predict performance in sports practice.

One of the problems faced when defining training policies as those described by our work is to clearly quantify the physical limits of effective training and overtraining, which are assumed to be related to aerobic and anaerobic thresholds. [Saastamoinen et al., 2004] defined a model to obtain the values of these parameters according to a fuzzy decision-making methodology that relies on the criteria of expert trainers.

Also, there are many factors that influence athletic performance in highly competitive sports.[Yang and Chen, 2012] developed a quantitative method for predicting the performance of university volleyball teams by taking into account a set of influential factors and performing Grey incidence analysis [Liu and Lin, 2006]. They aimed at providing useful decision support for volleyball teams during competition season. Nevertheless, their research does not focus on individual performance of players like this work, but on factors that affect the team as a whole.

7.2.2. AmI systems in the sport domain

Several context-aware applications for athletes' training have already been introduced in the literature. The focus of these works is on collecting data related to specific applications. However, in most systems, these data are not directly used to obtain real-time feedback but for external analysis by human coaches. Below we outline some works belonging to this group.

In MarathonNet [Pfisterer et al., 2006] a WSN monitored runners in marathon events. Sensors on runners collected data on HR, time, and location. This data were sent via base stations along the track to a central database, where they were analyzed off-line. The system in [Michahelles and Schiele, 2005] assisted profes-

sional skiers. Using accelerometers and force-sensing resistors, skiers obtained data about their motions and visualized it, along with video footage, once exercise had finished. The SESAME project [Hailes, 2006] investigated systems to enhance the performance of athletes. It focused on heterogeneous wireless sensing devices, rather than on decision making like SAETA. Another sport-oriented monitoring system prototype focused on sensing was presented by [Baca et al., 2010]. In the prototype athletes were equipped with sensors to gather several biometrics and a mobile phone to transmit data. This data could be examined on-line by a human coach to provide feedback to the athlete during training. Another WSN monitoring platform oriented to sports is REMOTE, developed by [Llosa et al., 2009]. This platform gathered real-time data to characterize rower performance and boat motions, but it did not return real-time feedback to athletes. The commercial product Team^{2}Pro [Polar, 2011] enables the recording and study of fitness data in real-time for up to 28 players.

Besides, there is extensive previous research on the analysis of action and movement (similarly to SAETA) in different sport fields, some of them based on video analysis and pattern recognition [Shao et al., 2014; Oikonomopoulos et al., 2005]. There are also many proposals that employ kinematic sensors. For example, [Watanabe and Hokari, 2006] and [Ghasemzadeh et al., 2009] developed systems to evaluate the quality of golf swings, from the data gathered by gyroscopic sensors attached to specific control points of the golfers. [Spelmezan and Borchers, 2008] introduced a sensor system to alert users immediately of incorrect motions and body positions. Their prototype used sensors attached to the human body and inserted into boots for detecting mistakes during snowboarding.

Recently, some works providing real-time feedback in sports have also appeared, such as the context-aware monitoring application by [Castillejo et al., 2013], which can be applied to suggest a series of exercises to improve the condition of the athlete.

7.3. System architecture

Figure 7.1 shows a schematic view of our training system. The sensing and communication infrastructure is based on the WSN architecture developed in previous work [Vales-Alonso et al., 2010]. This infrastructure is in charge of managing two types of data:

- A set of static data, with players' information that does not change during the training time, such as performance profile and age, an environmental data such as the training program or the height of the net.
- A set of real-time data comprising both ambient parameters such as temperature or humidity, and data associated to each player such as her HR or the elapsed training time.

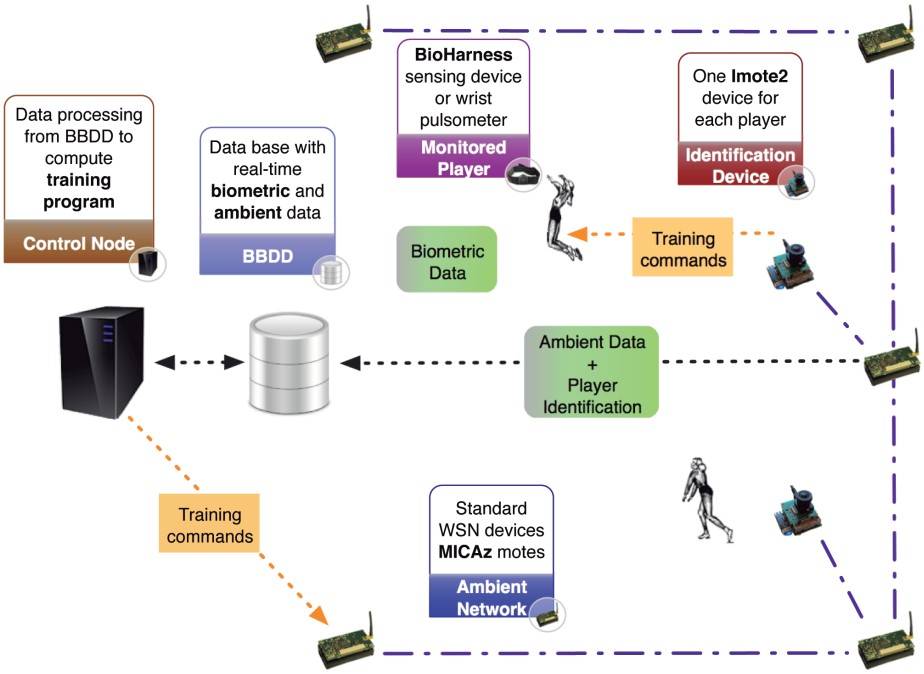


Figure 7.1: Global system architecture

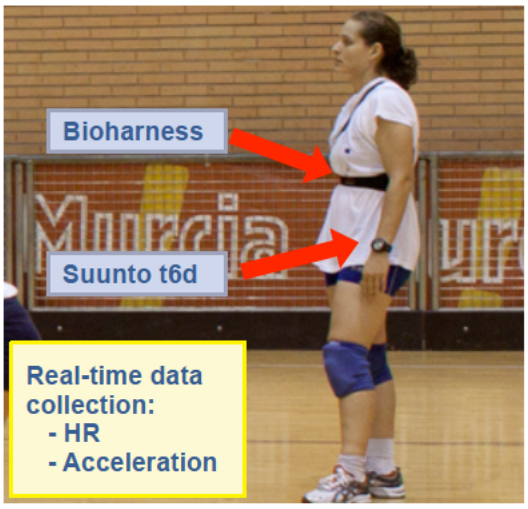
Our model considers a WSN deployed around the perimeter of the training field (see Figure 7.1) in charge of collecting ambient data and connecting users with the decision-making node (control node or CN). Similarly to the prototype developed in [Vales-Alonso et al., 2010], this *ambient* network has been implemented using standard WSN hardware (MICAz and Imote2 motes). Nevertheless, it has been adapted to the particular sensing and communications needs of SAETA.

Specifically, the player equipment consists of three devices:

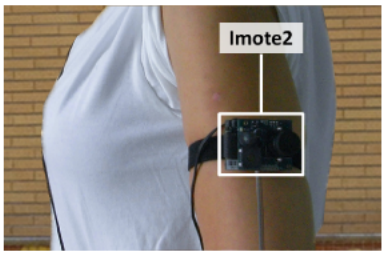
- *Suunto t6d Black Smoke*: it is a wrist HR monitor carried by the player, shown in Figure 7.2a. Besides, it keeps track of training zones and estimates post-exercise oxygen consumption (EPOC) and the maximum VO_2 consumption.

- *Imote2*: provides communication and sensing capabilities to the player. It includes a wireless transceiver to communicate with the network, and sensing elements to monitor player biometry. It also includes an acoustic interface to receive training orders, and it assists the player's localization procedure by responding to the localization messages from the control node. The device can be placed on the training field, close to the player or, as in the case of the BioHarness, the player may wear it as in Figure 7.2b.

- *Zephyr BioHarness*: the *belt-shaped* biometric sensing device shown in Figure 7.2c. It measures body acceleration and communicates this data through an embedded Bluetooth interface. The player carries the device fastened to her body.



(a) Player with equipment



(b) Imote2

(c) Zephyr BioHarness

Figure 7.2: SAETA biometric hardware

On the other hand, the CN provides the intelligence of the whole system. It coordinates sensing and communication devices, collects data from sensing devices, generates real-time training decisions and transmits them to the player. Its interface and the decision-making modules have been developed with Matlab. The interface, shown in Figure 7.3, permits to access all the functionality.

7.4. Effort control module

This section describes the customized match module of our training control system. This module has been chosen to tackle the coaches' need to adjust exercise intensity during a match [Moras Feliu, 1994]. It is important to remark that our system reacts to changing training conditions (of the user and the environment) in real time and provides training decisions for coaches and players. These decisions are computed by the CN (see Figure 7.1). The CN is able to infer intelligent decisions by using a knowledge base of data from previous training sessions.

Customized matches are standard competition matches, where the coach may vary game conditions to address a specific training objective. For example, a common goal in customized matches is to increase aerobic capacity by extending playing time and by reducing activity. The customized match module aims to complete a training program by controlling players' HR but by using an additional methodology to the one described in our previous works [Vales-Alonso et al., 2010] and [Vales-Alonso et al., 2012a]. The new module utilizes the numerical approximation of HR regimes to infer future values. Then, if the estimated HR is low, the control indicates an increase in activity, and *vice versa*.

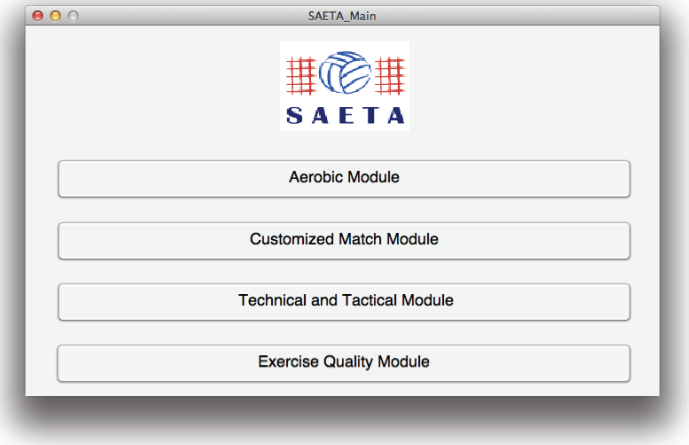


Figure 7.3: SAETA main window

Since our aim is to control a player's HR, we first need to characterize his/her performance. According to the type of training activity, HR performance can be classified into different intensity levels or classes (from Low to VMAX) [Haskell et al., 2007]. The HR in each class is in a percentage range of the recommended HR_{max}.

1. Low activity: <50%.
2. Moderate activity (Maintenance/Warm up): [50%,60%].
3. Weight Control (Fitness/Fat burn): [60%,70%].
4. Aerobic (Cardio training/Endurance): [70%,80%].
5. Anaerobic (Hardcore training): [80%,90%].
6. VMAX (Maximum effort): >90%.

An accurate formula to compute the maximal HR [Tanaka et al., 2001] for a healthy person follows the relationship $HR_{max} = 208 - 0.7 \times age$.

7.4.1. Decision-making process

For each player under control the coach selects total activity time and a training profile (TP). The TP is represented by a vector $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_6)$, where each position corresponds to an HR class, from low activity (ρ_1) to VMAX (ρ_6). Each position of $\boldsymbol{\rho}$ represents the ratio of training time that the player should accomplish in the corresponding HR class during the total activity time. Thus, the coach inputs TP values such that $\sum_{i=1}^{6} \rho_i = 1$. An example of this module input window is presented in Figure 7.4.

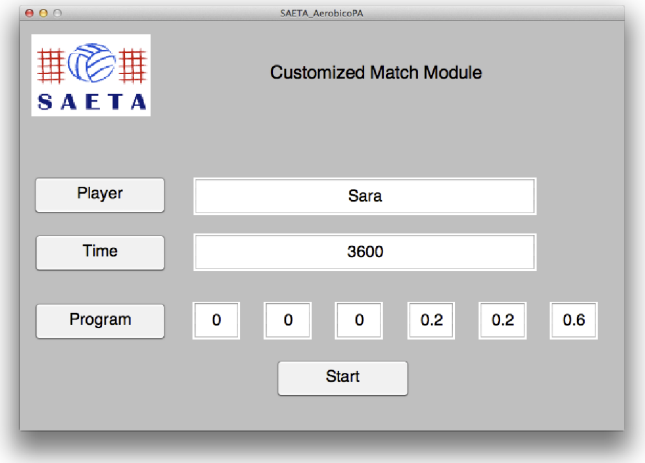


Figure 7.4: Input window for customized match module

Once the total activity time and TP are selected, the control module computes HR ranges (using the athlete's profile) and starts sampling the player's HR, classifying each sample in its range. Thus, after the jth sample the module now has the ratios in every HR range observed so far. These ratio values are utilized to compute predictions on the future evolution of the players' effort. Next, we discuss three different methods used for this task:

1. Best-fit polynomial approximation
2. Kalman filter
3. Sliding-Window distribution estimation

7.4.2. Best-fit polynomial approximation

The first approach was already discussed in [Vales-Alonso et al., 2013a]. It consists of using individual polynomials for each ratio $(p_i, i = 1, \ldots, 6)$, providing best-fit curves for the observed curves. To compute these polynomials, we have selected the least square method, which requires at least a number of points greater than the degree of the polynomial m. In our implementation, after the jth $(j > m)$ sample the algorithm utilizes the last $n > m$ samples, n being a parameter of our module program. With this estimation, the CN computes the difference between the selected TP and the estimated TP at each instant j, following the expression:

$$\Delta[j] = \sum_{i=1}^{6} \theta_i(\rho_i - p_i[j + \text{STEP}]) \tag{7.1}$$

Where $\theta = (\theta_1, \ldots, \theta_6)$ are weighting factors of the difference between the approximated values and those required by the coach. Besides, STEP is an algorithm

parameter that represents the estimation horizon in seconds (120 seconds in our tests). Thus, $p_i[j + \text{STEP}]$ is the value of polynomial i at instant j+STEP. For the conformity tests we have considered $\theta = (-1, -1, -1, 2, 2, 4)$. These weights balance the ratio of time that a player must train in each HR regime and can be modified, according to the athlete or coach's needs to fit training needs. Should the player's training decrease, these values will lead to negative Δ's, or positive in the contrary. Therefore, the output of this decision-making process indicates to players and coach whether the intensity of training must increase or decrease (respectively, positive or negative $\Delta[j]$ values). In addition, we have considered the importance of underlining the difference between estimated and required TP for the VMAX regime ($\theta_6 = 4$), due to the difficulty of achieving that exercise level. This weight stresses the trend of increasing training intensity if required to fulfill the selected TP.

7.4.3. Kalman filter

The second alternative that we have considered is the use of a predictor based on Kalman filter [Kalman, 1960], which allows us to solve, in a computationally efficient and recursively manner, least-squares estimation problems. These problems consist on estimating parameters by minimizing the squared discrepancies between observed data (the effort ratios in our work), on the one hand, and their expected values on the other.

The algorithm finds the least squares best estimate by going through data one by one. The filter gets the new best estimate for a given set of data once a new measurement is added by using, the new measurement, the old estimate, and some measure of confidence in the old estimate. As a result it provides an estimate of the state each time a new measurement is included. In order to define our Kalman filter, we must set the *state vector*. In our case contains the ρ vector values and its first derivative (velocity):

$$x = (\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6, \dot{\rho}_1, \dot{\rho}_2, \dot{\rho}_3, \dot{\rho}_4, \dot{\rho}_5, \dot{\rho}_6) \tag{7.2}$$

Then, the Kalman estimation proceeds iteratively with the next two steps (see next Figure), executed at each time instant j. Namely, a prediction and correction process called Kalman filter cycle:

1. *Prediction step.* The algorithm determines a state estimate $\hat{x}_j^-$ and an estimate for the error covariance P_j^- which provides an indication of the uncertainty associated with the current state estimate. The state-evolution equation is:

$$x_j^- = A\hat{x}_{j-1} + w_j \tag{7.3}$$

where

$$A = \begin{pmatrix} I_6 & \Delta t\ I_6 \\ 0 & I_6 \end{pmatrix} \tag{7.4}$$

is the state transition matrix, and w_j represents a normal distributed process noise sample with covariance matrix Q. Besides, in our case we have set the initial value $P = I_{12}$. In our model, empirical value $Q = 0$ gave us the best results.

2. *Correction step.* It provides a feedback by incorporating a new measurement value into the *a priori* estimate to get an improved *a posteriori* estimate. In this step the expected state $\hat{x}_j^-$ is corrected by following the equations provided in Figure 7.5. The main idea is relating the current state to a measurement z_j through a lineal model with a matrix H, and a normal distributed process noise sample v_j with covariance R. This is:

$$z_j = H\hat{x}_j^- + v_j \tag{7.5}$$

where

$$H = (I_6|0_{6x6}) \tag{7.6}$$

Afterwards the model corrects the estimation $\hat{x}_j^-$ by means of this measurement z_j. The empirical value $R = 0.1I_6$ gave us the best results. Besides, the Kalman gain K denotes that while the measurement error covariance R gets closer to zero, the measurement z_j is more trusted, while the predicted measurement is less trusted.

Therefore, we can estimate future ρ values if a set of previous measures is known (such us our HR ratio measurements). This estimation is updated every time a new measurement is obtained. To infer ratios with a prediction horizon of STEP stages, the Kalman prediction must be applied STEP times. Note that each estimated ρ value obtained by Kalman filter must be normalized. Moreover, as in the previous case of the polynomial approximations, once the expected ratio is computed, the difference between the selected TP and the estimated TP is obtained using expression 7.1. The same weighting coefficients have been used in our evaluation tests to correct the training offset.

7.4.4. Sliding-Window distribution estimation

Finally, in order to estimate the ratios of each effort regime for an athlete during a training session, a Sliding-Window distribution estimation method was also tested. The foundation of this last method is to assume that, in absence of external stimulus, the athlete will continue to perform with similar effort in the near future as he or she has done in the recent past.

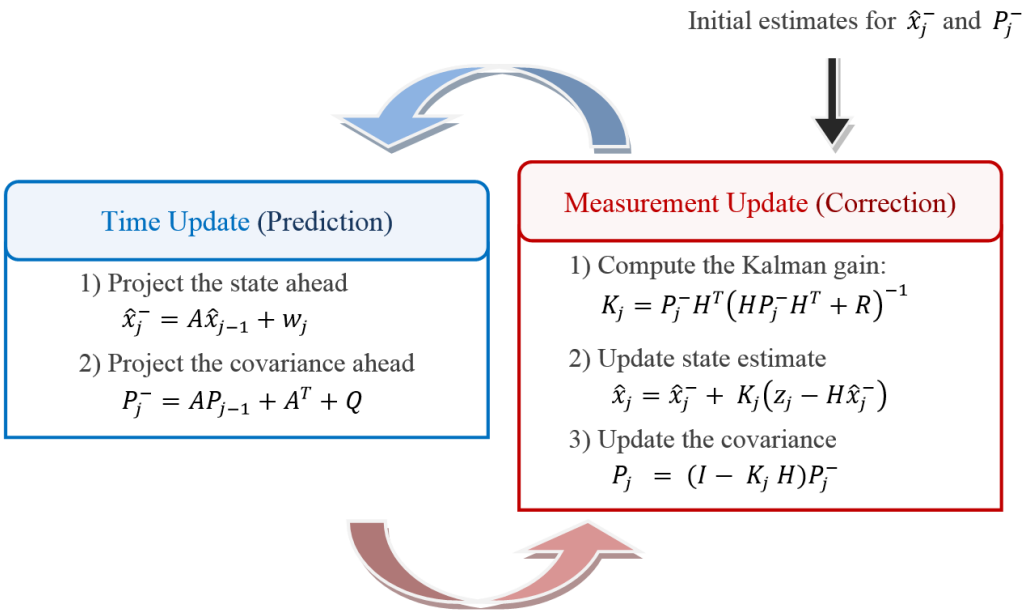


Figure 7.5: Kalman filter cycle

To do this, let us assume that $r_i[j]$ is a discrete sequence of the values zero and one, where one means that the heart rate of the athlete sampled at time j lies within the ith HR defined range, and zero means that the heart rate is outside this range. At any given time j, the fraction of time an athlete is within each HR range is simply calculated as the fraction of accumulated HR samples in each range over the total of HR samples:

$$\rho_i[j] = \frac{\sum_{\tau=1}^{j} r_i[\tau]}{\sum_{i'=1}^{6}\sum_{\tau=1}^{j} r_{i'}[j]} \tag{7.7}$$

If we define the accumulated value of HR samples within each range as $\mathrm{HR}_i[j] = \sum_{\tau=1}^{j} r_i[\tau]$, thus Eq. 7.7 can be simplified as:

$$\rho_i[j] = \frac{\mathrm{HR}_i[j]}{\sum_{i'=1}^{6} \mathrm{HR}_{i'}[j]} \tag{7.8}$$

After a number of new samples (defined by STEP in our case), the updated ρ_i values must be between the ranges specified in the next equation. This is, between zero and STEP new values have been added to the ith HR range, and exactly STEP new samples have been added to the total:

$$\frac{\mathrm{HR}_i[j]}{\mathrm{STEP} + \sum_{i'=1}^{6} \mathrm{HR}_{i'}[j]} \leq \rho_i[j + \mathrm{STEP}] \leq \frac{\mathrm{STEP} + \mathrm{HR}_i[j]}{\mathrm{STEP} + \sum_{i'=1}^{6} \mathrm{HR}_{i'}[j]} \tag{7.9}$$

None of the external factors that can affect athletes' heart rate will be taken into account for the forecast method described in this section so, as stated before, we will assume that the athlete will keep a similar intensity in the near future exercise performance as he or she has kept in the recent past. This is, if we look at the most recent n samples of HR, the prediction made by this method will be to forecast that the next STEP samples will be classified following the same distribution or, to put it in another words, if $\Delta \mathrm{HR}_i^n[j] = \mathrm{HR}_i[j] - \mathrm{HR}_i[j-n]$ new samples have been added to the ith HR range within the last n samples, each one of the next STEP samples has an independent probability of $\frac{\Delta \mathrm{HR}_i^n[j]}{n}$ of being classified within the ith HR range.

Taking this approximation, and working only with the expected value for the random variables involved, the ratios for each HR range can be expressed as shown in the following equation:

$$\hat{\rho}_i[j] = \frac{\mathrm{HR}_i[j] + \frac{\mathrm{STEP}}{n}\Delta \mathrm{HR}_i^n[j]}{\mathrm{STEP} + \sum_{i'=1}^{6} \mathrm{HR}_{i'}[j]} \tag{7.10}$$

This formulation of the predicted ρ_i ratios can be updated with each new sampled HR value, and also ensures that the sum of all predicted $\rho_i[j]$ for any given j is always one, as in the Kalman filter case. Finally, the difference between the expected and the required TP are obtained also with expression 7.1.

7.4.5. Validation

We have tested this module in a real customized match training session of the UCAM volleyball team. The training objective was to augment the effort in relation to a standard match. To that end, the coach eliminated serves and idle periods and immediately passed the ball when the training was interrupted. During training the players' biometrics were monitored in real time using the BioHarness system. An example of samples collected from two different players is represented in Figure 7.6. We can observe how the different roles of player's in a match can produce different HR profiles. Specifically in our tests, the physical effort of a hitter produces higher HR frequency samples than that of a setter. Hitters therefore require different TPs and indications. By using the customized match module, the coach is able to adapt training to each player.

In order to validate our methodology, we used an external feedback signal to the player's HR, which is controlled by the program module output. A rise in training intensity points to an increase in the player's HR modulated by the output of the program. This process is similar to the result obtained when a real player receives a command to increase training. The same procedure is followed when the decision is to decrease training intensity. Thus, we modulate the HR of a player by adding a feedback signal γ.

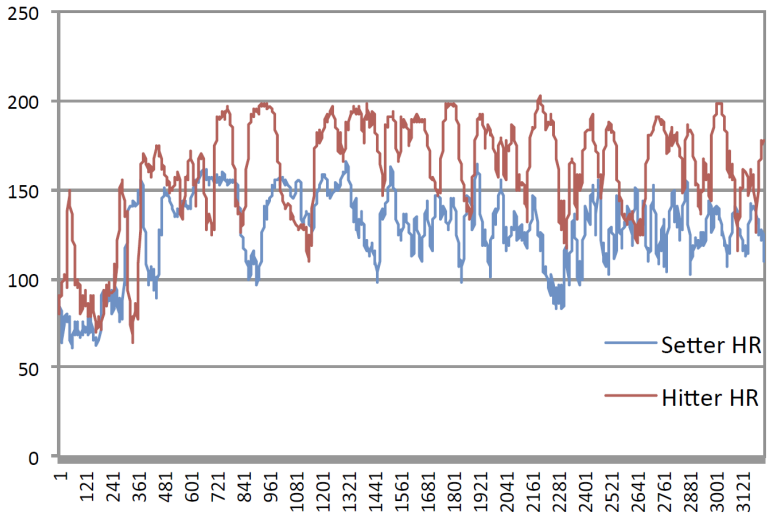


Figure 7.6: Real-time HR samples during a customized match training

The computation is done as follows:

$$hr'(j) = hr(j) + \gamma(j) \tag{7.11}$$

where γ is the output of the single pole IIR system,

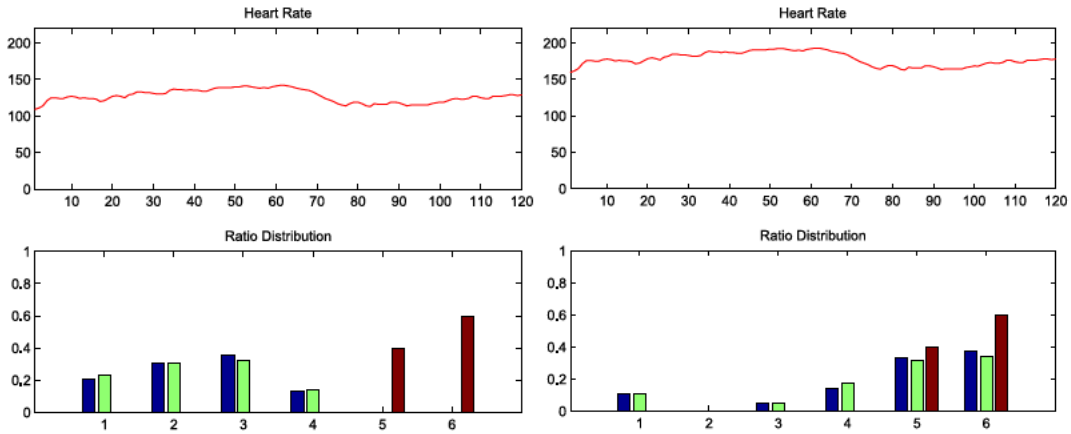
$$\gamma(j) = a\gamma(j-1) + \Delta[j] \tag{7.12}$$

and weight a serves to adjust the strength of the feedback signal. In our validation tests we have selected $a = 0.99$.

Finally, we computed the difference between the TP required by the coach and the real training performed. This difference is computed in two forms: with and without HR correction (i.e. without using the control module). Our system will be validated if the use of the feedback signal brings the real TP closer to the TP configured by the coach. The difference between both TPs shows the goodness of fit of our methodology.

We have tested our customized match module with two players: a setter and a hitter and two different effort distributions (ρ). Figure 7.7 shows an example of the output (see [Vales-Alonso et al., 2013a] for the full set of results). For each HR class, the following three data ratios are shown:

- The blue bar represents the estimated HR values computed with p_i polynomials and STEP=120
- The green bar represents the current HR of the player
- The brown bar represents the required ratio



(a) No feedback (b) Feedback

Figure 7.7: Real samples of hitter heart rate and their distribution in intensity classes under training time objective $\rho = (0, 0, 0, 0, 0.4, 0.6)$

In all cases the results show how the difference between objective TP (brown bar) and real training obtained (green bar) is greatly reduced by using the feedback signal. Clearly, introducing a feedback during the training session leads to a better match with the selected TP.

If the other methods (the Kalman filters or the Sliding-Window distribution estimation) are used, similar trends have been observed (but with a different final distribution). Next we compare the performance of the three methods.

Figure 7.8 shows comparatively the error (measured as norm 2, i.e., the root square of the squared difference between the expected and the real ratio) in the prediction achieved by the three methods for a complete training session of 1 hour (from 1 to 3600 seconds). The curves depicted show the predicted ratio with a horizon of STEP=180 seconds. Note that predictions starts in 300 seconds since we require collecting enough data to start predictions, and we have set this time as 300 seconds in our experiments. Moreover, the Sliding-Window method also uses a window of $n = 300$ samples. Clearly the polynomial approximation has as side effect large oscillations, as can be expected. Comparatively, both the Kalman and the Sliding-Window approximations have a better performance without oscillations. Although in some cases the Kalman filter prediction surpasses in quality the Sliding-Window method (e.g. for ρ_4 around the 600 seconds), the common trend is that the Sliding-Window method achieves less prediction error for that horizon STEP.

Another important issue is how STEP should be selected. On the one hand, if STEP is too small the predictions will be better but rather useless, since only real data is used to decide the training orders (the feedback). For example, if STEP is null this is similar to a coach that has been collecting statistics of a player during training and decides the next training based on that statistics. What our

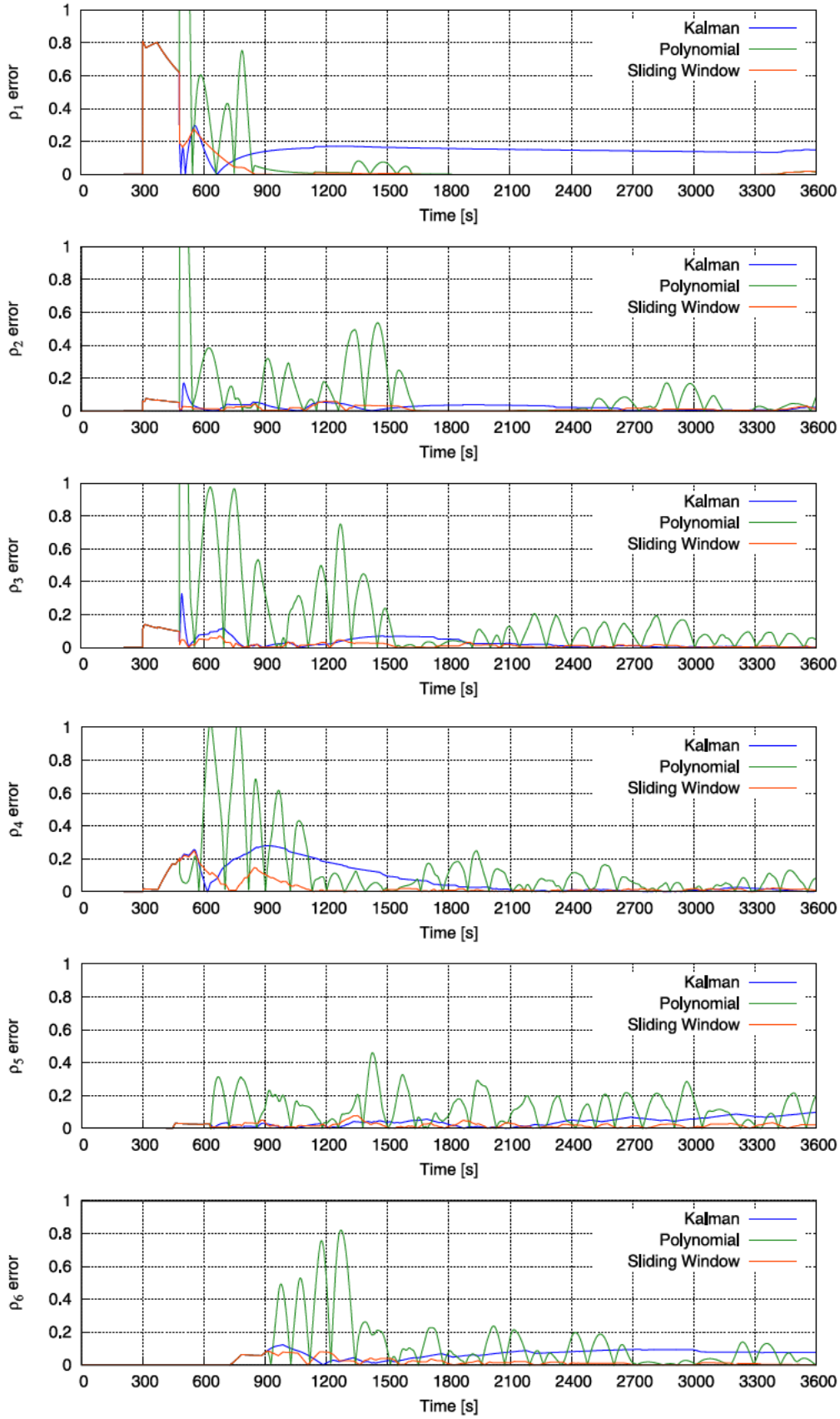


Figure 7.8: Estimation error

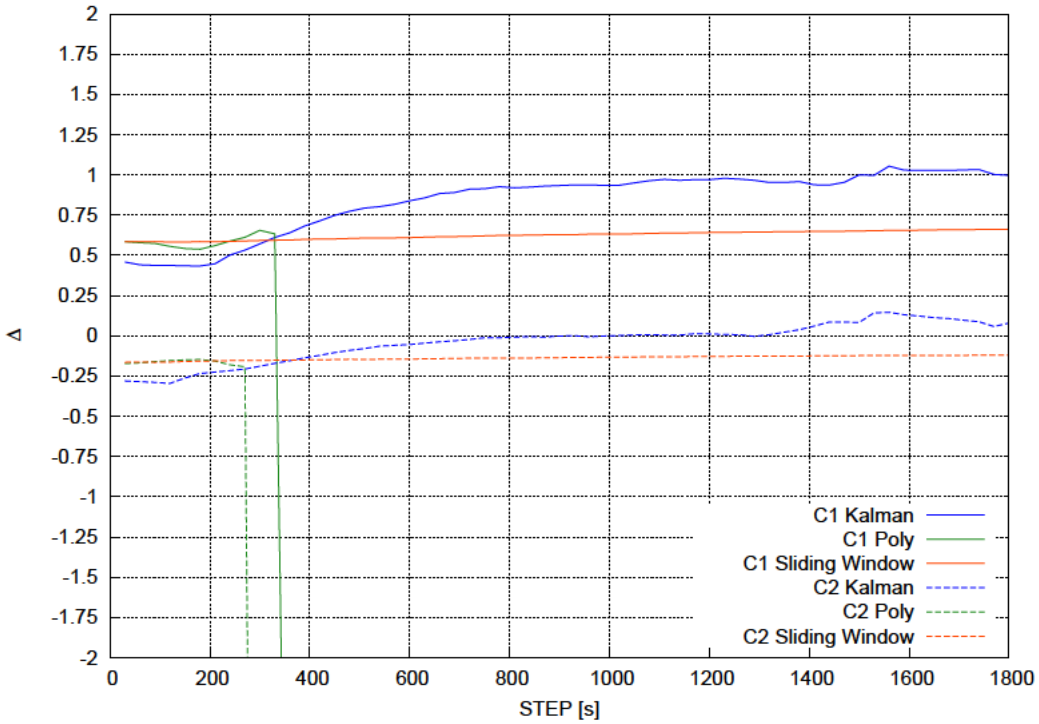


Figure 7.9: Final Δ *versus* STEP value, player 1

system allows is to infer what will happen in the future and add this information to the decision making process. On the other hand, when STEP is too large the prediction process will become unreliable, due to the dynamic evolution of the HR regime ratios. Therefore, the feedback obtained is likely to be incorrect.

Summarizing, there must be a balance in the STEP value. An experiment has been performed to show this. Figures 7.9 and 7.10 depict the final Δ value (i.e. Δ at 3600 seconds instant) computed trough expression 7.1 for both players and both example TPs. As being stated previously, this value indicates the quality of the training, being better for values closer to 0. As shown in Figs. 6 even using feedback is difficult to reach the most intensive HR regimes since the TP is too homogeneous, and therefore the final Δ tends to be positive. Besides, for the second TP, and since it concentrates in the hardest regimes, the feedback is more effective and the final results are closer to zero and even negative, due to some over-effort.

As shown in the Figures 7.9 and 7.10, even though the polynomial approximation achieves a reasonable performance for STEP values below 300 seconds, beyond this point this method becomes unusable, with a stepper performance drop. For the player 1, the setter, the best training for the first TP is obtained for STEP values in the order of 150 seconds, using the Kalman filter. In the case of considering the second TP, the best training is obtained again with the Kalman filter for a broad values of STEP around 1000 seconds. In the case of the second player, for both

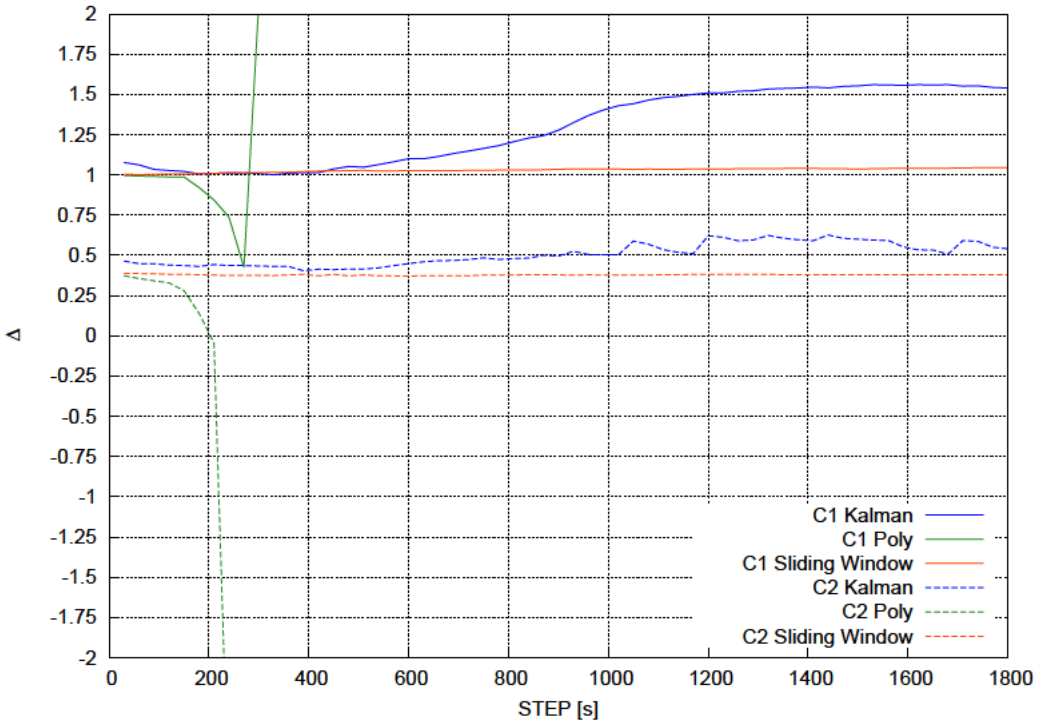


Figure 7.10: Final Δ *versus* STEP value, player 2

TPs the best method is the polynomial approximation, for 300 and 200 seconds respectively. However, we must note that *a priori* is not possible to known the best step, and that in this case the Sliding-Window methods is safer, since, as show in these Figs., the response is almost independent from the STEP parameter.

As conclusions, if a careful preliminary study can be performed for a given player to obtain a suitable prediction methodology and parametrization, benefits can be obtained in the fine-tuning of the athletes' training. Otherwise, both the Kalman and Sliding-Window methods are worth being considered. A trend has been observed in the results where Kalman filter obtains similar or better results than the Sliding-Window technique for STEP values around 300 seconds.

7.5. Technical-tactical control system

Volleyball matches, among other competitive sports like tennis or basketball, are characterized by the interleaving of short periods of intense physical activity and resting intervals. In order to improve players performance in real matches, this interleaving is often reproduced in training sessions, in the so-called technical-tactical or repetition exercises. For example, Figure 7.11 shows a coach supervising a hitter training. It consists of several *stages* starting with an *activity period* where the player hits balls either from the coach or from another player, and followed by

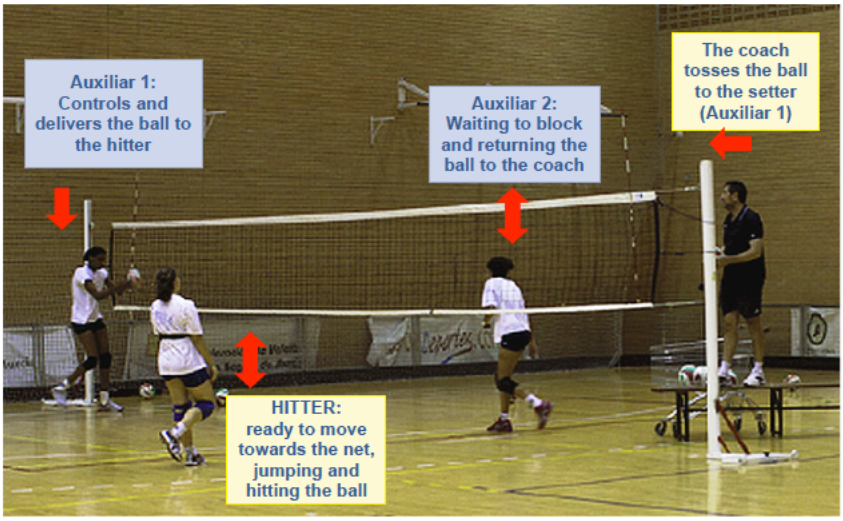


Figure 7.11: UCAM Volley Murcia hitter repetition exercises

a longer *resting period*, as recommended by physical training authors (e.g. [Moras and Zurita, 1999; Baechle et al., 2000]). An adequate control of the effort intensity is critical for the training success: a minimum effort (above a Lower threshold) is needed to trigger an adaptive physical response, but excessive effort (above an Upper threshold) could lead to over-training and fatigue, thus reducing training benefits [Billat et al., 1999].

There are several methods to obtain an estimation of the effort level of an athlete during training [Bunc et al., 1995], such as measuring lactic acid accumulation in the blood stream, measuring oxygen consumption in the respiratory activity, monitoring HR, and so forth. The most accurate one is the measurement of lactic acid, while the others are indirect mechanisms that try to estimate it. However, since lactic acid measurement is an invasive method it cannot be used regularly during training sessions. Consequently, the HR is broadly selected as a good estimator of the effort level [Cooper, 1991], and is used as such in this work.

Effort control during technical-tactical sessions is challenging even for coaches, since predicting dynamic HR evolution is difficult due to the stochastic nature of HR signals (see previous section). Actually, most coaches select static rest and activity periods and, at most, analyze HR information off-line. Moreover, although physical performance is extremely important, it is not the only requirement in technical-tactical training. The correct execution of the exercises is also of paramount importance for the success of the training sessions. However, only under constant supervision by the coaches it can be implemented, since the body motions and the intensity applied should be monitored. In this case, a good indicator could be the use of accelerometers that provide information about one or several body part movements.

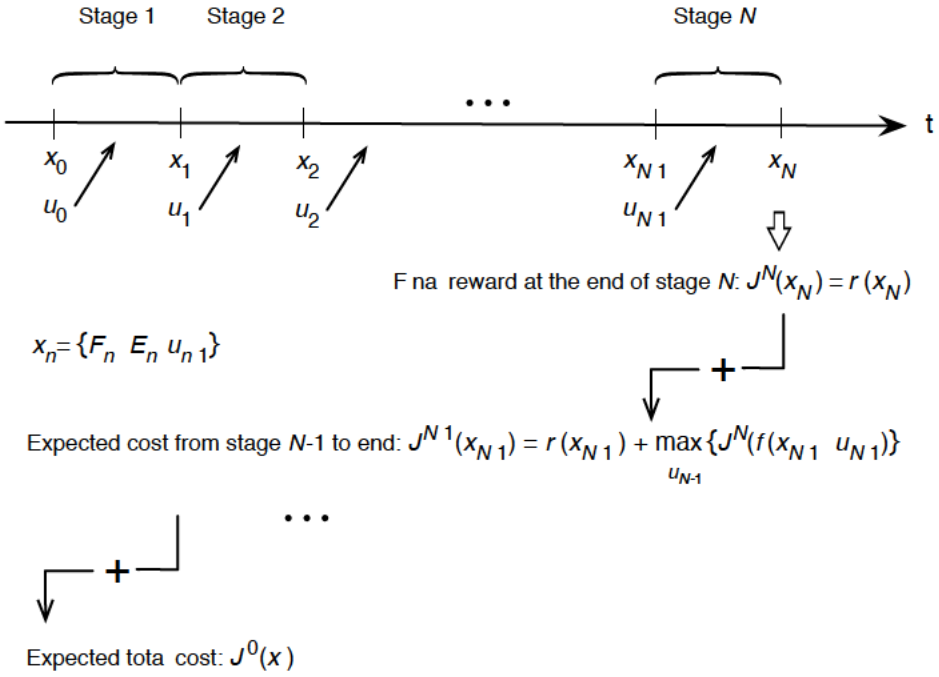


Figure 7.12: Graphical representation of Bellman´s equation

7.5.1. Dynamic Programming model

A good training control should find a combination of activity and rest durations to help players keep a high HR (and so, effort) while avoiding over-training, as described in the introduction. This problem, rather than static, has a strong dynamic component since both activity and rest durations must be decided per stage, and will depend on the HR evolution during exercise. The mathematical branch addressing optimal decision making in dynamic time discrete systems (*i.e.* those evolving at a numerable set of time instants) is Dynamic Programming (DP). This section describes the DP formulation used in our system. DP has been successfully applied to a wide variety of problems, ranging from computational biology [Zhang and Wong, 1997] to bioinformatics [Li et al., 2009] or path planning [Patek et al., 1999]. However, as far as we know the use of DP in the sport domain is a novel contribution of this work.

The foundation of any dynamic program is Bellman's equation [Bellman and Dreyfus, 1962], which describes the expected maximal reward from stage n onwards, $J^n(\cdot)$, as the sum of rewards from simpler subproblems associated to each stage n, $n+1$, etc. Bellman's equation can be expressed as:

$$J^n(x_n) = \max_{u_n}\{\mathbb{E}\left[r(x_n) + J^{n+1}(f_n(x_n, u_n))\right]\} \tag{7.13}$$

In order to explain Bellman's equation, let us introduce its elements:

- A training session consists of a sequence of N stages. Each stage includes one activity period of duration t_{acti} and one rest period of duration t_{rest}.
- Each stage in the session is denoted by an integer n in the set $\{1, \ldots, N\}$.
- Besides, at the end of each stage n the HR data must be mapped to quantitative levels of effort E_n and fatigue F_n to describe the physical response of the athlete. Section 7.5.2.1 describes how they can be computed.
- With the information available at the end of stage n, the system selects a control (decision) denoted by u_n. This control is applied and has effect in stage $n+1$. The control states the duration of the activity (t_{acti}) and resting (t_{rest}). In order to simplify the problem, each of these durations have been limited to a pair of possible values, namely *short* and *long* durations[1]. As a consequence, the control applied in each stage n can be fully described by an index between 0 and 3, with the mapping provided in Table 7.1.
- The state of the system during stage n is denoted by x_n, and must include relevant information to make the decision u_n at the end of each stage n. In this particular problem, x_n contains the level of effort and fatigue, as well as the control selected to current stage u_{n-1}. That is, $x_n = (E_n, F_n, u_{n-1})$.
- A reward function, denoted by $r(x_n)$, which represents the reward obtained in state x_n, and is described in Section 7.5.2.2. Besides, in this model the reward is assumed to be deterministic, so $\mathbb{E}\left[r\left(x_n\right)\right]$ will be expressed as $r(x_n)$ henceforth.
- A transition function $f_n(x_n, u_n)$, which computes the next state of the system (x_{n+1}) that would be reached if control u_n is applied in state x_n at the end of stage n.
- Finally, the value function $J^n(x_n)$, the maximal expected reward that can be obtained from stage n to the final stage N.

Given this notation (see also Figure 7.12) it is possible to explain recursive Eq. (7.13): it states that the value function associated to the player's training from stage n onwards, $J^n(x_n)$, can be computed as the sum of the reward expected at stage n plus the expected value function from the next stage onwards, $J^{n+1}(x_{n+1})$.

Then, the goal is to solve Eq. (7.13), that is, to find the optimal control u_n for each state x_n. That is, select u_n depending on n, on the control applied before

[1]The use of more levels in the control is straightforward mathematically but would require a very large set of training examples (see Section 7.5.2.3), which may be unfeasible in practice, so it has not been considered.

Control value u_n	**Activity duration** $t_{\mathbf{acti}}$	**Resting duration** $t_{\mathbf{rest}}$
0	short (20")	short (60")
1	short (20")	long (90")
2	long (30")	short (60")
3	long (30")	long (90")

Table 7.1: Possible values for the control parameter u_n

u_{n-1}, and on the effort and fatigue levels E_n, F_n. The result of the optimization process is a *policy* in the form of a *look-up* table, which maps the optimal control for each system state.

7.5.1.1. Markov Decision Process formulation

In order to solve the previous DP, the key element is describing the transition function $f(\cdot)$. However, given a state x_n and a selected control value u_n, the next state of the system x_{n+1} is not completely determined but stochastic. To overcome this limitation our system has been modelled as a Markov Decision Process (MDP), in which the evolution between states is assumed to be governed by a Markov process regulated by the selected control. That is, we assume transition probabilities $p^{(n)}_{x_n,x_{n+1}}(u_n)$: the probability of evolving from state x_n to state x_{n+1} if control u_n is selected at end of stage n (*i.e.* applied in stage $n+1$). Note that different stages or controls may have different transition matrices associated. Thus, Eq. (7.13) can be rewritten as:

$$J^n(x_n) = r(x_n) + \max_{u_n}\{\sum_{\forall j} p^{(n)}_{x_n,j}(u_n)\, J^{n+1}(j)\} \tag{7.14}$$

We can provide a more convenient matrix-notation of the last expression, denoting as $\mathbf{J^n}$ and $\mathbf{r}$ the vectors containing in their ith element the function values and rewards associated to the state with ordinal i (see note[2]), and as $\mathbf{P^{(n)}}(u_n)$ the transition matrix of the Markov process at stage n. Thus, the previous equation is transformed to:

$$\mathbf{J^n} = \mathbf{r} + \max_{u_n}\{(\mathbf{P^{(n)}}(u_n)\,\mathbf{J^{n+1}}\} \tag{7.15}$$

Since the problem has a finite horizon of N stages, the value function can be computed directly from the previous expression (see Figure 7.12). The computing process starts at the end of last stage N, selecting $\mathbf{J^N} = \mathbf{r}$ (at the last stage only he reward associated to the level of fatigue and effort reached must be added to the value function). Then the process continues backwards, deciding the best controls

[2]Given a state $x = (E, F, u)$ it is univocally associated to the ordinal $i{=}E{+}F{\cdot}\mathrm{MAX}{+}u{\cdot}\mathrm{MAX}^2$, *e.g.* state (2,1,3) has ordinal 54. With our configuration MAX=4, the number of possible controls, as well as the number of effort and fatigue levels (see Section 7.5.2.1).

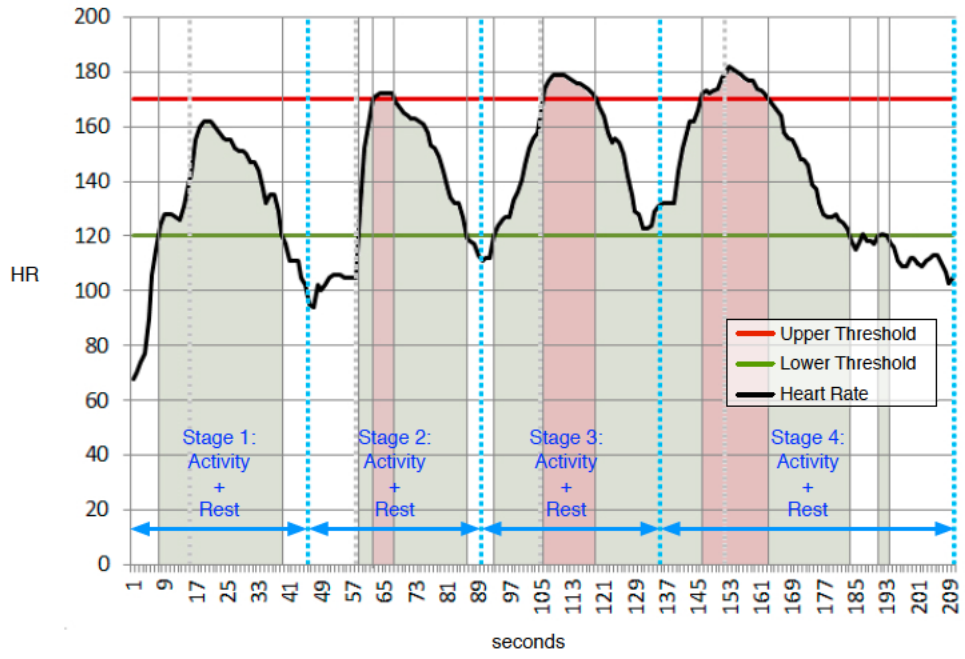


Figure 7.13: Evolution of the HR of a player during a training session as a function of time. Effective effort and over-training areas are highlighted

to apply at stage N (u_{N-1}), $N-1$ (u_{N-2}), and so on. At each stage n the value function $\mathbf{J^{n+1}}$ is already known from the previous optimization step, and decisions result from merely selecting, among the possible controls, the one that maximizes the sum of current reward plus $\mathbf{J^{n+1}}$. This process continues until eventually stage 1 is reached, where the state x_0 and control u_0 are assumed to be fixed (see next section) and the total value function J^0 is computed, and the look-up tables can be computed as well (as described in next section).

7.5.2. Technical-tactical effort control module

In this section we provide precise details about how the technical-tactical effort control module addresses the computation of the effort and fatigue level, and how they are used to represent a suitable reward function that achieves the goal of maximizing player's effort, while avoiding fatigue. Besides, details about the MDP matrix computations and the evaluation of this method are provided as well.

7.5.2.1. Effort and Fatigue level computation

Figure 7.13 exemplifies the evolution of the HR of a player as a function of time during one session. Typically, HR increases during activity periods and decreases during resting periods. Note that the session consisted of four stages in this example, thus N=4. Additionally, Figure 7.13 shows the two thresholds described at

the introduction (Lower and Upper thresholds), which will be respectively denoted as $\mathrm{HR}_{\min}$ and $\mathrm{HR}_{\max}$ henceforth. For each individual, the value of $\mathrm{HR}_{\max}$ can be estimated based on her/his age following several methods [Robergs and Landwehr, 2002], being $\mathrm{HR}_{\max} = 205.8 - (0.685 \cdot age)$ an accurate estimator. The value of $\mathrm{HR}_{\min}$ is usually set between 60%-70% of $\mathrm{HR}_{\max}$, according to cardiology studies [Kindermann et al., 1979]. Let us remark that both thresholds ($\mathrm{HR}_{\min}$, $\mathrm{HR}_{\max}$) can be customized in training for each player, according to their specific needs and to the competition calendar, *e.g.* reducing $\mathrm{HR}_{\max}$ to prevent fatigue before critical season events.

As stated previously, the HR data at each stage should be used to obtain levels of effort and fatigue corresponding to the physical performance of the athlete. These levels should be related to:

1. The time either in fatigue (HR> $\mathrm{HR}_{\max}$) or in effort regime ($\mathrm{HR}_{\min} \leq$ HR $\leq \mathrm{HR}_{\max}$).

2. Fatigue intensity, which is higher as the HR increases.

3. Effort intensity, which is higher as the HR approaches to $\mathrm{HR}_{\max}$ without surpassing it.

All these conditions are met if the areas above and below the respective HR thresholds are considered (see Figure 7.13). The higher the corresponding area, the higher the levels F_n or E_n. Formally, for stage n both areas ($a_{F,n}$, $a_{E,n}$) can be computed respectively as:

$$a_{F,n} = \sum_{t=T_{n-1}+1}^{T_n} \mathrm{HR}[t] 1_{\{\mathrm{HR}_{\max} < \mathrm{HR}[t]\}} \tag{7.16}$$

$$a_{E,n} = \sum_{t=T_{n-1}+1}^{T_n} \mathrm{HR}[t] 1_{\{\mathrm{HR}_{\min} \leq \mathrm{HR}[t] \leq \mathrm{HR}_{\max}\}} \tag{7.17}$$

Where T_n denotes the last time instant belonging to stage n, HR[t] is the HR sample at time t, and $1_{\{condition\}}$ is an indicator function, having the value 1 if condition is fulfilled and the value 0 otherwise. Finally, by quantizing these areas it is possible to obtain respectively the fatigue (F) and effort (E) levels. To do so, let us define MAX as the number of quantitative levels, and respective threshold areas $a_F^0 = 0, a_F^1, \ldots, a_F^{\mathrm{MAX}} = \infty$ for fatigue, and $a_E^0 = 0, a_E^1, \ldots, a_E^{\mathrm{MAX}} = \infty$ for effort. Thus, F_n is given by i if $a_{F,n}$ belongs to the interval $[a_F^i, a_F^{i+1})$, with i ranging from the lowest intensity level 0 to MAX-1. The quantified index of effort (E_n) is obtained similarly. The parameter MAX has been set to 4 due to practical reasons (see Section 7.5.2.3). Area thresholds are input parameters that must be defined by the coaches and can be customized to particular players. In our tests,

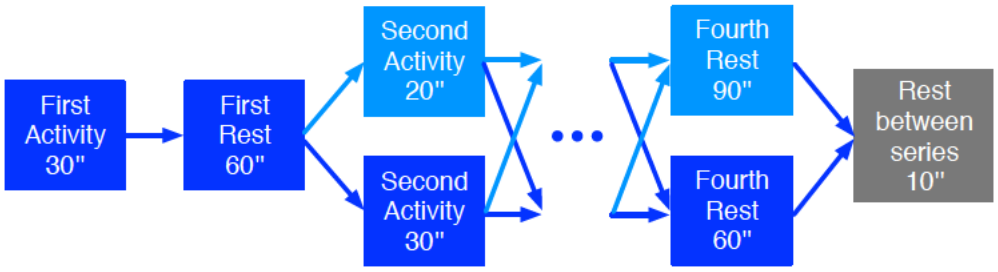


Figure 7.14: Configuration scheme of a technical-tactical training session

area limits were set by dividing the possible area values in equally spaced intervals. The maximum and minimum values were set empirically from the HR recorded from each player. Note that this is an arbitrary criterion decided from the data analysis, but others could have been selected as well.

7.5.2.2. Reward function

In the DP model is critical to define a suitable reward function $r(x_n)$. The value of this function must be higher if the effort made by the athlete during the stage is higher, but a penalty shall be applied if the fatigue threshold has been reached. For this goal we have selected the next linear reward function:

$$r(x_n) = -\alpha F_n + \beta E_n \tag{7.18}$$

The influence of effort and fatigue levels is weighted in Eq. (7.18) by α and β, which are customizable parameters of the system. For the validation tests performed in Section 7.5.2.4 they were $\alpha = 5$ and $\beta = 3$ according to the coaches recommendations. With these values, a negative reward indicates over-training, whereas a positive reward indicates that the exercise has been executed mostly within the specified margins of effort. Let us remark that the ratio $\frac{\beta}{\alpha}$ is controlling the fatigue penalty. The higher it is, the more reactive will the system be against fatigue. Note that different reward functions than formula (7.18) could have been used as well, provided that they meet the requirements above.

7.5.2.3. Experimental computation of transition matrices

To build Eq. (7.15) we need to compute the transition matrices $P^{(n)}(u_n)$ for each stage n and control u_n. In practice, they have been determined by evaluating the relative frequencies of the transition events obtained experimentally, following the next procedure for each player:

- During a series of technical-tactical training sessions of UCAM Volley Murcia, the player was asked to perform training using all the alternative configurations paths shown in Figure 7.14. This way, all possible control combinations

have been traversed. Note in Figure 7.14 that the first stage is always performed with the same control u_0 (configuration 2 of Table 7.1), since at the beginning of the exercise there is no further information to select a suitable control, and it must be selected with another criterion. During technical-tactical training sessions this first stage was predefined by the coaches as a static physical warm-up phase, therefore this is reproduced in our model by fixing the first stage, and associating no reward to it (hence, value function $\mathbf{J^0} = \mathbf{J^1}$). After this static first stage, in the exercises considered there were 3 additional stages (since N=4). Thus, for each exercise session there were 8 possible path combinations.

- For each exercise session, the effort and fatigue levels for each stage n (E_n, F_n) were determined from her HR signal. Since the control applied u_{n-1} is known, the state x_n can be registered.
- The number of transitions between states x_n and x_{n+1} were determined $(\eta^n_{x_n,x_{n+1}}(u_n))$, as a function of u_n.
- This experiment was repeated 10 times (*i.e.* a total of 80 exercise repetitions including all path combinations) to increase the data available to construct the transition matrices.
- Finally, the relative frequency $(\frac{\eta^n_{x_n,x_{n+1}}(u_n)}{10})$ was computed leading to matrices $\mathbf{P}^n(u_n)$.

In our experiments, data was recorded in a period of approximately one month, performing between 2 and 4 exercise series per training day, which is intense but possible for elite players training. If the number of levels of effort and fatigue, or the number of stages should be increased, we would require a very large set of training examples (physical exercise repetitions) to determine the transition matrices using the previous procedure, which could be prohibitive. In this situation, some alternative model like NARMAX [Cooper, 1991] customized to each player could be considered.

7.5.2.4. Evaluation and practical use

Usually, coaches establish a fixed control sequence for training sessions, *i.e.* static durations for all the activity and resting phases. Thus, in order to evaluate the performance of the selected DP method, the optimal solution has been compared to the possible fixed policies that coaches may apply (see Table 7.1):

- *easy* policy - A short activity duration and a long rest duration.
- *hard* policy - A long activity duration and a short rest duration.

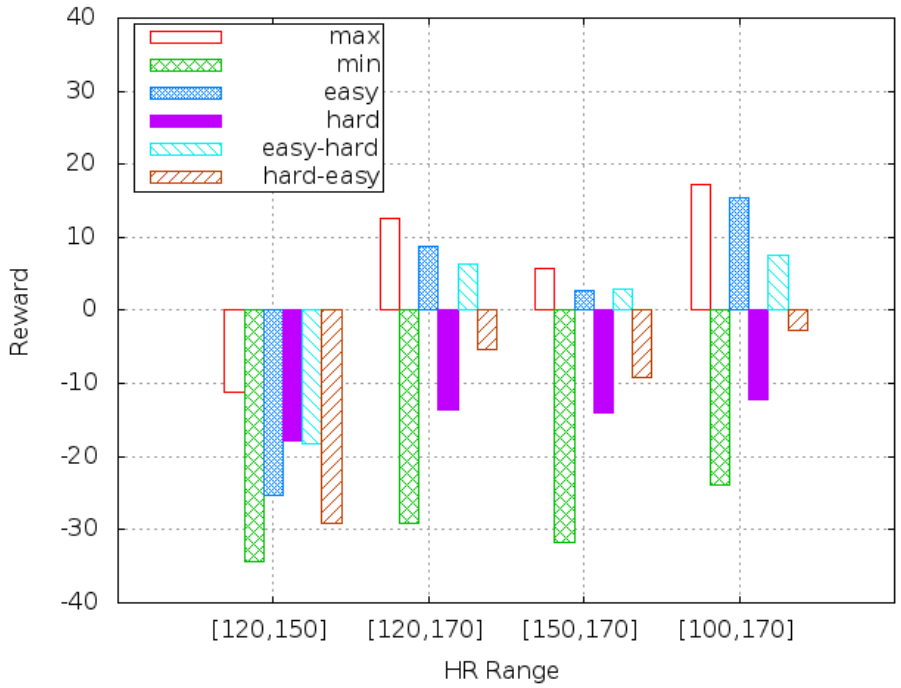


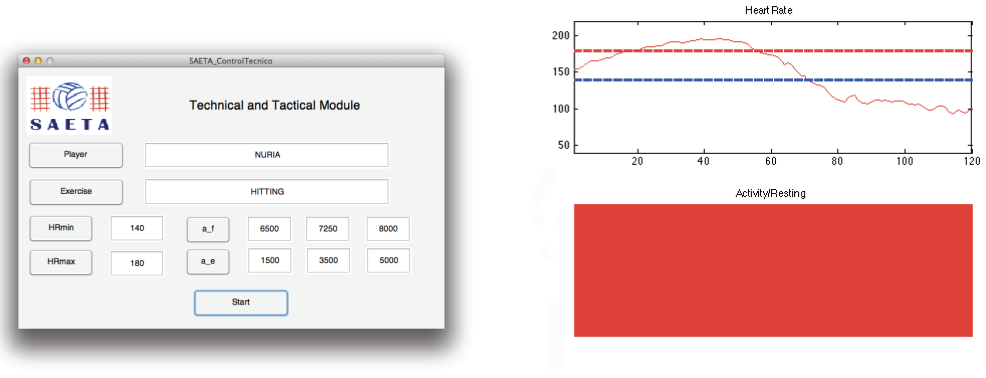
Figure 7.15: Total value function, J^0, obtained under optimal selection policy (*max*) versus several suboptimal policies (*min*, *easy*, *hard*, *easy-hard*, *hard-easy*) in a technical-tactical training session

- *easy-hard* policy - A short activity duration and a short rest duration.
- *hard-easy* policy - A long activity duration and a long rest duration.

Figure 7.15 shows the total value function, $\mathbf{J}^0$, achieved for each of the aforementioned policies for different $\mathrm{HR}_{\min}$ and $\mathrm{HR}_{\max}$ thresholds (similar to those suggested by [Moras Feliu, 1994]). In all cases, as stated before, the initial control selected was u_0=2 (long activity, short resting). Besides, in Figure 7.15 the optimal policy is also shown and labelled as *max*, while the worst possible (*min*) policy[3] has also been included as reference of the lower limit for the result. According to these results, the proposed policy outperforms all the others, as expected. The *easy* policy obtains usually a better score than the *hard* policy, since it avoids the negative effect of over-training, but it achieves a lower rate of effort. The same effect occurs when the *easy-hard* policy is compared against the *hard-easy* policy. Moreover, the results clearly vary with the HR thresholds, fatigue being determinant when $\mathrm{HR}_{\max}$ is lower. In some configurations, *e.g.* 150-170, the expected maximal reward doubles the best fixed configuration (easy policy in this case).

Figure 7.16 shows the control window that the SAETA technical-tactical module offers to the coaches. They must select the player and customize the HR thresholds,

[3]Computed by substituting max operator with min operator in eq. (7.15)



(a) Input (b) Output

Figure 7.16: Technical-tactical module input and output windows

as well as the area limits to quantize effort and fatigue. Then, the module obtains the HR signal, and displays a green screen for the coach if the player must perform activity, or a red one when the player must rest. The player is also notified with acoustic signals through the Imote2. The optimal policy is stored in a lookup table built as explained in Section 7.5.2.3, which is consulted in real-time by the technical-tactical module. Table 7.2 shows a partial view of the computed lookup table obtained in our tests. For example, for the HR range of 150-170, if $x_1 = (F = 0, E = 0, u_0 = 2)$, which corresponds to the third row and third column in Table 7.2, the best policy would select a control u_1=3 (long activity and long resting) for the second stage.

Finally, let us remark how the method developed at this section can be easily customized to different season epochs or training constraints by adapting the limits of HR rates. As shown in Figure 7.15 these variables clearly determine the rewards, and so affect the training orders. For example, in Table 7.2 if $x_1 = (F = 0, E = 1, u_0 = 1)$ for HR limits 120-150 (i.e. first column, second row) the control to apply is u_1=2 (long activity and short resting), whereas with HR limits 100-170 (i.e. fourth column, third row) a control u_1=1 (short activity and long resting) would be applied. This last situation could correspond for example with the first part of the season where physical tone must be acquired, and effort starts at a lower HR.

7.6. Exercise quality control module

In the previous section, the effort control method has been introduced. As discussed before in Section 7.5, the correct execution of the exercises is also of great importance in technical-tactical sessions. Therefore, an additional module of SAETA has been developed in order to control the quality and intensity of specific movements of the players.

$HR_{min} = 120$ $HR_{max} = 150$	$HR_{min} = 120$ $HR_{max} = 170$	$HR_{min} = 150$ $HR_{max} = 170$	$HR_{min} = 100$ $HR_{max} = 170$
{0,1,0,2,1}	{0,1,0,2,1}	{0,0,0,2,2}	{0,1,0,2,1}
{0,1,1,1,2}	{0,1,1,1,2}	{0,0,1,1,2}	{0,1,1,0,1}
{0,1,2,1,1}	{0,1,1,3,3}	{0,0,1,2,3}	{0,1,1,1,1}
{1,1,0,2,3}	{0,1,2,0,1}	{0,0,1,3,3}	{0,1,1,2,3}
{1,1,1,1,2}	{0,1,2,1,1}	{0,0,2,0,3}	{0,1,2,0,3}
...	...	...	...

Table 7.2: Examples of optimal policies for the HR ranges considered in Fig. 7.15. The policies are shown as a set $\{F_n, E_n, n, u_{n-1}, u_n\}$. Only the first 5 rules are shown for each range

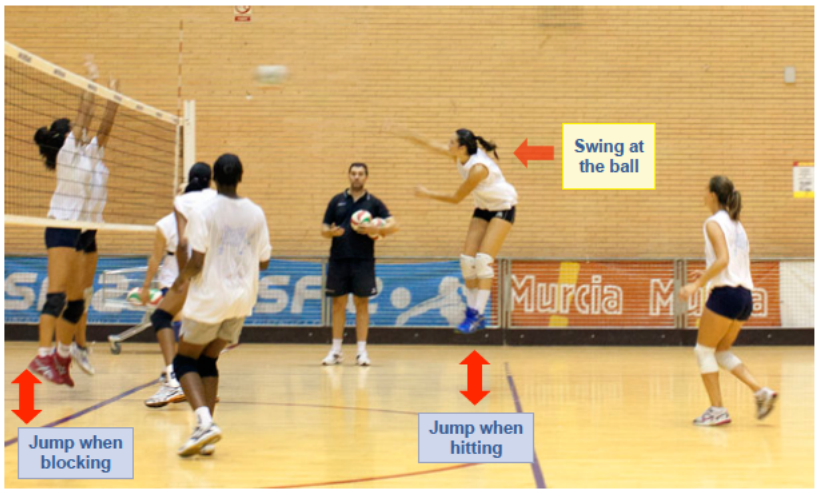


Figure 7.17: Different jump actions

This module has been focused to jumping since it is one of the most critical movements in volleyball, which is involved in most offensive and defensive actions. Hence, SAETA has taken this movement as the target example. This is a complex task, due to the wide range of movements that share similar attributes with jumping, but also due to the variety provided by the physical characteristics of different players. As an example, Figure 7.17 shows two different jumping actions: defensive blocking, in which the player must rise her arms to block the incoming ball; and attack hitting, in which the player must turn the body to transfer more energy to the ball.

The sensing architecture presented in Section 7.3 obtains 3-axis acceleration data with the help of the BioHarness sensing belt attached to the players' body. Note that the belt always orientates the acceleration axes in the same way, therefore simplifying the classification and study of the player's movements.

Belt data is sent on real-time to a control module in charge of:

- Detecting the type of movement performed by the player with a k-NN (k Nearest Neighbors) classifier [Fix and Hodges Jr, 1951].
- Rating movement intensity by means of a histogram analysis technique.

7.6.1. Jump detection using k-NN classifier

The k-NN classifier requires a previously labelled set of example data that is used to predict the class of new samples using a majority vote mechanism. Despite its simplicity, it is adequate for our real-time movement detection problem. In the proposed solution, the following features have been selected:

- Acceleration peak in the X, Y and Z axes (features number 1 to 3 respectively).
- Ratio between the acceleration peak and the average of the 6 samples before and after the peak for the X, Y and Z axes (features number 4 to 6 respectively).

The k-NN classifier was cross-validated against a record of training sessions containing blocking jump and reception-setting actions. Samples of the Bioharness' 3-axis accelerometer are taken each second and labelled manually by a supervising technician as jump and no-jump actions. These samples were taken both during training sessions including only repetitions of the same action and also during simulated matches. A subset of the former ones is shown in Figure 7.18, where the first 210 seconds represent 20 repetitions of jump actions and the rest of the samples correspond to 20 repetitions of receive-setting actions. The complete record of labelled samples used for training and validation purposes contains around 150 jump actions and 150 non-jump actions in total.

During cross-validation experiment, a random set containing the 95% of the samples in this record was used for training the k-NN classifier, while the remaining 5% were used to test the accuracy of the selected method and tune its parameters. This experiment was repeated 30 times, and the classification ratio results averaged.

The parameters for the classifier are: (a) the set of features considered, and (b) the number of neighbours taken into account for the k-NN majority vote (k). Table 7.3 summarizes the average classification ratio results (jump detection ratio or, in other words, classification true positives) of the cross-validation, for k ranging from 1 to 6, and several feature sets (labelled as F). As shown in this table, the best results (highlighted in bold font) correspond to k=2, with feature sets $\{1, 2, 3, 5, 6\}$ and $\{1, 2, 3, 5\}$. Moreover, with this parameters the ratio of true negatives was 100% (i.e., all no-jumping actions were correctly classified).

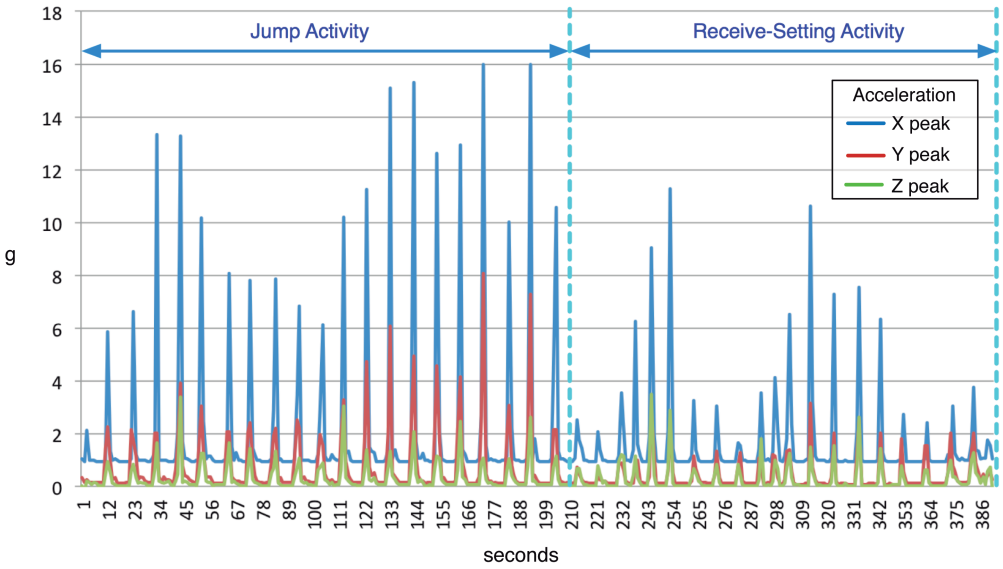


Figure 7.18: Partial record of labelled jumping/non-jumping data during training sessions. Note that g=9.8 $\frac{\mathrm{m}}{\mathrm{s}^2}$

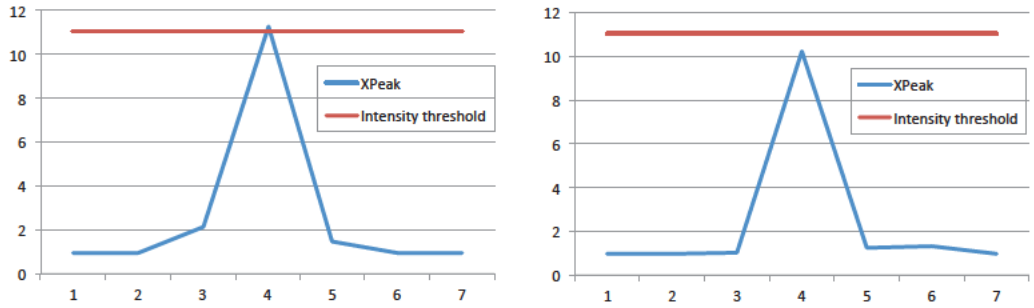
	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
$F=\{1,2,3\}$	0.7857	0.7381	0.7857	0.7619	0.7857	0.7381
$F=\{1,2,3,4,5,6\}$	0.8333	0.8095	0.8810	0.8095	0.8810	0.8571
$F=\{1,2,3,4\}$	0.7143	0.6429	0.6905	0.7143	0.7143	0.5952
$F=\{1,2,3,5\}$	0.9048	**0.9286**	0.8810	0.8810	0.8333	0.8333
$F=\{1,2,3,6\}$	0.8333	0.8810	0.8571	0.8333	0.7857	0.7857
$F=\{1,2,3,5,6\}$	0.8571	**0.9286**	0.8810	0.8810	0.8571	0.8333
$F=\{1,2,3,4,6\}$	0.8333	0.8095	0.8571	0.8095	0.8095	0.8095
$F=\{1,2,3,4,5\}$	0.8810	0.8095	0.8333	0.8333	0.8095	0.8095

Table 7.3: True positives for the classifier for several k values and feature sets (F)

7.6.2. Jump intensity analysis

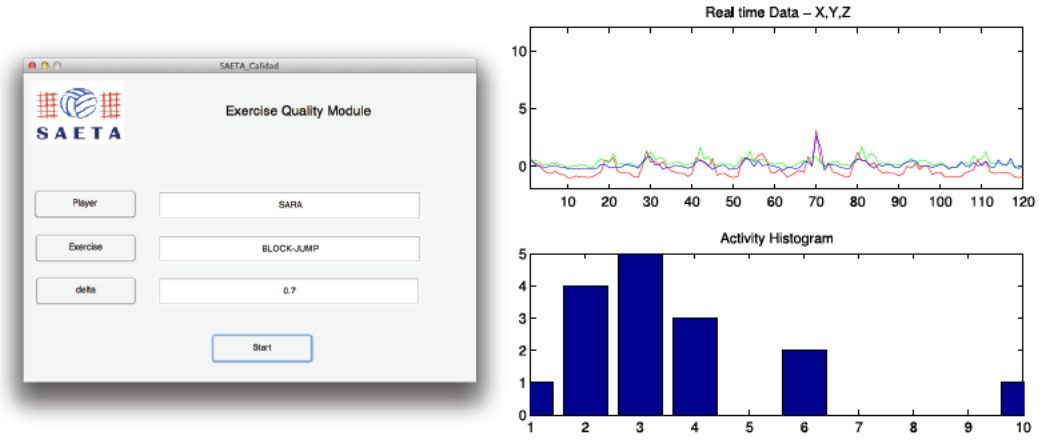
Once the detected movement is classified as a jump, the methodology dictates that it must be labelled as valid or invalid according to its intensity and the player profile. In order to do this, the acceleration peak value has been selected as a good estimation of the intensity of the jump. This value is then compared to a threshold to check its validity.

Figure 7.19 shows an example of two executions of a jump, one of them valid and the other not valid. The value of the threshold depends on the physical attributes of the player performing the jump. In the proposed methodology, this threshold is computed by taking into account previous jumps of the same player. The jump is considered valid if the intensity is above the intensity level of $100\delta\%$ of the previous jumps (being δ a parameter customized by the coaches between 0 and 1). Therefore, increasing δ leads to more intensity imposed to the player. To determine the minimum intensity level, a histogram of jump intensities is stored for each player



(a) Valid jump action (b) Not valid jump action

Figure 7.19: Peak X acceleration value for valid (a) and not valid (b) jump actions



(a) Input (b) Output

Figure 7.20: Quality module input and output windows

based on data from previous 5 sessions. The number of sessions is limited since physical performance depends on training and season epoch and adding old data may distort the results.

Finally, the information about detected jump actions and their validity is shown to the coaches in real-time with the help of graphics like the ones in Figure 7.20. If the jump is correctly performed and its intensity is above the defined threshold, the players are also notified through their wearable device. In these graphics, the real-time data for X, Y and Z axes acceleration is shown in the upper plot, while the lower plot shows a histogram of the intensity of jumps.

7.7. Conclusions

The new SAETA AmI system for professional team sport training has been developed in this work. The main focus was on developing suitable decision-making systems for different aspects of players training. Unlike some of the previous related works, the SAETA system tries to cover several facets of elite players training, including both performance and technique, providing automated real-time feedback based on coaches' expertise and players' physical status.

Effort control module analyze real-time training HR data in order to estimate the ratios of each effort regime based on several methodologies: best-fit polynomial approximations 7.4.2, Kalman filter 7.4.3 and Sliding-Window distribution 7.4.4. In our tests, depending on the STEP value, some methodologies obtain better predictions than others. For STEP values beyond 300 seconds, polynomial approximation becomes unusable, while Kalman and Sliding-Window methods are worth being considered.

For technical-tactical training, a supervised-learning module based on Markov Decision Processes computes the best configurations (activity and resting durations) for repetitional exercises. As shown in Figure 7.15 the performance expected for a player increases notably in comparison with common training policies where these durations are static, and it also avoids fatigue when possible, contrary to some static policies. Besides, a module to analyze exercise quality has been also developed to complement the technical-tactical module. The quality module is able to detect player jumps and evaluate their intensity with respect to the history of actions of the player. The detection process was based on a k-NN algorithm, and it achieved a classification ratio of nearly 93% for true-positives and 100% for true-negatives, depending on the k parameter and the features selected, as shown in Table 7.3. Validation tests performed with the *UCAM Murcia* volley team have demonstrated the utility of the SAETA system for both coaches and athletes.

Although training context-aware applications have been proposed before in the academic literature, to our knowledge none of them has deal with both effort and quality control, providing real-time decision feedback to both coaches and players. The application of dynamic programming based on Markov Decision Processes to this type of problem is also an innovative approach that can lead to new tools for elite sport training, but also for amateurs trying to increase their performance.

Further research efforts might improve our training system by augmenting the details of the information reported to coaches, and the development of new decision-making modules. In addition, predetermined parameters, like the reward function coefficients or the effort and fatigue area thresholds, could be dynamically determined from the analysis of realtime data instead.

Conclusions and future work

This is the nature of war.
By protecting others,
you save yourselves.

We have proposed some novel optimization models applied to RFID, WSN and AmI field, including solving strategies for these models. In this part, we summarize the main results of this doctoral thesis and outline the future lines of research.

Optimization in RFID dense-reader environments

In Chapter 1 we introduced a novel optimal scheduler for a particular dense reader environment composed by several readers which must share a single frequency channel. The scheduler proposed exceeds in performance to heuristic algorithms, improving the average number of tags identified in an RFID facility. Besides, the effect of the reading protocols has also been studied in depth, concluding that a dynamic FSA algorithm excels static frame length ones. Indeed, the impact of using knowledge about tag population in the scheduler has been analyzed. It has been concluded that, even assuming errors in the estimation, our scheduler is able to obtain a higher performance than a reference model where the average population is perfectly known.

We aim at extending this scheduler to multi-channel scenarios, developing a model that allow full resource reutilization. It is discussed in Chapter 2, where we introduce OSL, a general multi-channel scheduler for dense reader environments.

OSL is formulated to obtain maximal space resource reutilization and to avoid mutually interfering readers operating simultaneously on the same frequency channel. A practical solver to tackle the optimization problem is implemented by means of a parallel simulated annealing. The solutions obtained can be used to implement a system based on look-up tables computed off-line and, more important in our opinion, establishes the performance limits achievable. Therefore, results should be useful for evaluation of real-time heuristics. As a future work we aim at developing optimal schedulers with support of fairness criteria and priority among readers.

Another critical topic is studying the dynamics of such systems.

We have determined analytically in Chapter 3 the mean number of identifications from a population of several tags within a bounded time interval. This method can be easily adapted to any frame-length selection mechanism, and allows characterizing FSA systems. As demonstrated, readings depend on the initial population and there are non-linear effects which have been neglected before in RFID works.

As future work we will move forward on developing formulations for continuous-time schedulers and proposing commercial products based on them.

WSN Deployment

In deterministic deployment conditions, we have proposed in Chapter 4 a gunshot location procedure based a Pareto front approach to optimize large scale deployments. The location procedure is based on hyperbolic multilateration using data from a synchronized sensor network. We also propose a practical distributed synchronization algorithm for that purpose, with low energy consumption. Sensor network planning follows a Pareto front approach, using a monotone descent method without derivatives that is compatible with realistic optimization functions. Our results are clearly superior to random placement, achieving a 50% cost reduction for 85% coverage. In the two-dimensional scenario, event detection by at least four nodes is required to reach a satisfactory gunshot location accuracy.

In a first approach to the field of random deployment, in Chapter 5 we have developed mathematical and algorithmic procedures to calculate the best placement options for WSN Gaussian clusters in an area of non-homogeneous importance. In the formulation of the optimization problem we have captured the main characteristics of the WSN model and taken realistic assumptions: the clusters must be normally distributed over the monitored area and must also cover the best areas. At the same time, they must be connected, i.e. nodes cannot be isolated.

We have shown that the approximations selected in our analytical procedure achieve notably good results, closely resembling those obtained in experimental evaluations. Besides, as an important conclusion, in a single cluster optimal deployment hundreds (if not thousands) of sensors are required to completely sense some relative “small” regions. Iterative multi-deployment proves to be an effective strategy to overcome this problem. Results demonstrate how large non-homogeneous regions may be covered with several clusters using suitable configurations. The solution to the optimal placement problem developed in this work provides such configurations and largely outperforms previous heuristics.

In Chapter 6 we add some new important realistic features not considered in earlier works, like the use of realistic propagation and 3D surface models, the consideration of feasible node release mechanisms, the computation of propellant masses associated to the transport of the nodes, and the use of mass and lifetime con-

straints. Results shown that accounting for these aspects produce solutions radically different than those achieved with oversimplified previous models. As future work we will explore new optimization tools like the ant colony optimization or genetic algorithms, and will study new deployment scenarios like plane seeding or war and disaster zones where other optimization goals could be of interest.

AmI system for personalized sport training

SAETA, an AmI system for professional team sport training has been developed in Chapter 7. The main focus was on developing suitable decision-making systems for different aspects of players training. Unlike some of the previous related works, the SAETA system tries to cover several facets of elite players training, including both performance and technique, providing automated real-time feedback based on coaches' expertise and players' physical status.

SAETA decision engine is divided into three modules: Effort control module, technical-tactical control module and the exercise quality control module.

Effort control module analyze real-time training HR data in order to estimate the ratios of each effort regime based on several methodologies: best-fit polynomial approximations, Kalman filter and Sliding-Window distribution. In our tests, depending on the STEP value, some methodologies obtain better predictions than others. For STEP values beyond 300 seconds, polynomial approximation becomes unusable, while Kalman and Sliding-Window methods are worth being considered.

For technical-tactical training, a supervised-learning module based on Markov Decision Processes computes the best configurations (activity and resting durations) for repetitional exercises. The performance expected for a player increases notably in comparison with common training policies where these durations are static, and it also avoids fatigue when possible, contrary to some static policies.

The quality control module complement is able to detect player jumps and evaluate their intensity with respect to the history of actions of the player. The detection process was based on a k-NN algorithm, and it achieved a classification ratio of nearly 93% for true-positives and 100% for true-negatives, depending on the k parameter and the features selected. Validation tests performed with the *UCAM Murcia* volley team have demonstrated the utility of the SAETA system for both coaches and athletes.

Although training context-aware applications have been proposed before in the academic literature, to our knowledge none of them has deal with both effort and quality control, providing real-time decision feedback to both coaches and players.

In the near term we expect to develop new sport assistants based on foremost machine learning algorithms, like approximate dynamic programming. In the long term we wish to export this knowledge to other AmI fields such as Ambient Assisted Living.

Bibliography

– *Leone 5-10*
Frank Leone

Abramson, N. Packet switching with satellites. *Proc. National Computer Conference*, pages 695–702, 1973.

Abu-Mahfouz, A. and Hancke, G. Distance Bounding: A Practical Security Solution for Real-Time Location Systems. *Industrial Informatics, IEEE Transactions on*, vol. 9(1), pages 16–27, 2013. ISSN 1551-3203.

Acampora, G., Cook, D., Rashidi, P. and Vasilakos, A. A Survey on Ambient Intelligence in Healthcare. *Proceedings of the IEEE*, vol. 101(12), pages 2470–2494, 2013. ISSN 0018-9219.

Aitsaadi, N., Achir, N., Boussetta, K. and Pujolle, G. Multi-Objective WSN Deployment: Quality of Monitoring, Connectivity and Lifetime. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–6. IEEE, 2010.

Alcaraz, J., Vales-Alonso, J. and Garcia-Haro, J. RFID Reader Scheduling for Reliable Identification of Moving Tags. *Automation Science and Engineering, IEEE Transactions on*, vol. 10(3), pages 816–828, 2013. ISSN 1545-5955.

Ammari, H. and Das, S. Integrated coverage and connectivity in wireless sensor networks: A two-dimensional percolation problem. *IEEE Transactions on Computers*, vol. 57(10), pages 1423–1434, 2008.

Anderson, H. and McGeehan, J. Optimizing microcell base station locations using simulated annealing techniques. In *Vehicular Technology Conference, 1994 IEEE 44th*, pages 858–862. IEEE, 1994.

Ashton, K. That 'Internet of Things' Thing, in the real world things matter more than ideas. In *RFID journal*. 2009.

BACA, A., KORNFEIND, P., PREUSCHL, E., BICHLER, S., TAMPIER, M. and NOVATCHKOV, H. A Server-Based Mobile Coaching System. *Sensors*, vol. 10(12), pages 10640–10662, 2010. ISSN 1424-8220.

BAECHLE, T., EARLE, R. and (U.S.), N. S. . C. A. *Essentials of Strength Training and Conditioning*. Human Kinetics, 2000. ISBN 9780736000895.

BANNACH, D., AMFT, O., KUNZE, K., HEINZ, E., TROSTER, G. and LUKOWICZ, P. Waving Real Hand Gestures Recorded by Wearable Motion Sensors to a Virtual Car and Driver in a Mixed-Reality Parking Game. In *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, pages 32–39. 2007.

BARGER, T., BROWN, D. and ALWAN, M. Health-status monitoring through analysis of behavioral patterns. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35(1), pages 22–27, 2005. ISSN 1083-4427.

BARI, A., CHEN, Y., ROY, D., JAEKEL, A. and BANDYOPADHYAY, S. Designing hierarchical sensor networks with mobile data collectors. *Pervasive and Mobile Computing*, vol. 7(1), pages 128–139, 2011.

BARI, A., JAEKEL, A. and BANDYOPADHYAY, S. Clustering strategies for improving the lifetime of two-tiered sensor networks. *Computer Communications*, vol. 31(14), pages 3451–3459, 2008.

BELLMAN, R. and DREYFUS, S. *Applied Dynamic Programming*. Princeton University Press, 1962.

BILLAT, V., FLECHET, B., PETIT, B., MURIAUX, G., KORALSZTEIN, J. ET AL. Interval training at VO_{2max}: effects on aerobic performance and overtraining markers. *Medicine and Science in Sports and Exercise*, vol. 31, pages 156–163, 1999.

BROOKE, A., KENDRICK, D., MEERAUS, A., RAMAN, R. and AMERICA, U. The General Algebraic Modeling System. *GAMS Development Corporation*, 1998.

BUCUR, V. *Acoustics of wood*, vol. 1431. Springer Verlag, 2006.

BUEHLER, M., ANDERSON, R., SESHADRI, S. and SCHAAP, M. Prospecting for in situ resources on the Moon and Mars using wheel-based sensors. In *Aerospace Conference, 2005 IEEE*, pages 607–616. 2005.

BUEHLER, M., BOSTIC, H., CHIN, K., MCCANN, T., KEYMEULEN, D., ANDERSON, R., SESHADRI, S. and SCHAAP, M. Electrical properties cup (EPC) for characterizing water content of Martian and lunar soils. In *Aerospace Conference, 2006 IEEE*, pages 18–pp. IEEE, 2006.

BUENO-DELGADO, M. and VALES-ALONSO, J. On the optimal frame-length configuration on real passive RFID systems. 2011.

BUENO-DELGADO, M., VALES-ALONSO, J., ANGERER, C. and RUPP, M. A comparative study of RFID schedulers in dense reader environments. *Paper presented in IEEE international conference on industrial technology*, 2010.

BUENO-DELGADO, M., VALES-ALONSO, J. and GONZALEZ-CASTANO, F. Analysis of DFSA anti-collision protocols in passive RFID environments. *Paper presented in proceedings of the 35th international conference of the IEEE Industrial Electronics Society*, 2009.

BUNC, V., HOFMANN, P., LEITNER, H. and GAISL, G. Verification of the heart rate threshold. *European journal of applied physiology and occupational physiology*, vol. 70(3), pages 263–269, 1995.

CALVERT, T., BANISTER, E. W., SAVAGE, M. V. and BACH, T. A Systems Model of the Effects of Training on Physical Performance. *Systems, Man and Cybernetics, IEEE Transactions on*, vol. SMC-6(2), pages 94–102, 1976. ISSN 0018-9472.

CAO, X., CHENG, P., CHEN, J. and SUN, Y. An Online Optimization Approach for Control and Communication Codesign in Networked Cyber-Physical Systems. *Industrial Informatics, IEEE Transactions on*, vol. 9(1), pages 439–450, 2013. ISSN 1551-3203.

CASTILLEJO, P., MARTÍNEZ, J.-F., RODRÍGUEZ-MOLINA, J. and CUERVA, A. Integration of wearable devices in a wireless sensor network for an E-health application. *Wireless Communications, IEEE*, vol. 20(4), pages 38–49, 2013. ISSN 1536-1284.

CHEN, W.-T. An Accurate Tag Estimate Method for Improving the Performance of an RFID Anticollision Algorithm Based on Dynamic Frame Length ALOHA. *Automation Science and Engineering, IEEE Transactions on*, vol. 6(1), pages 9–15, 2009. ISSN 1545-5955.

CHOI, J. and LEE, C. An MILP-based cross-layer optimization for a multi-reader arbitration in the UHF RFID system. Sensors. vol. 11(3), pages 2347–2368, 2011.

CHUI-YU, C., CHENG-HSIN, K. and CHEN, K. Optimal RFID networks scheduling using genetic algorithm and swarm intelligence. *Paper presented in IEEE international conference on systems, man and cybernetics*, 2009.

CHUNG, I.-H., YEN, M.-C. and HWANG, C.-K. An accurate analytical formula for the essential joint probability of framed slotted aloha protocols. *Journal of the Franklin Institute*, vol. 350(10), pages 3432 – 3440, 2013. ISSN 0016-0032.

CLEMENTINE. Clementine project information. 2011.

COOPER, R. System identification of human performance models. *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 21(1), pages 244–252, 1991. ISSN 0018-9472.

DASGUPTA, K., KUKREJA, M. and KALPAKIS, K. Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks. In *Computers and Communication, 2003.(ISCC 2003). Proceedings. Eighth IEEE International Symposium on*, pages 341–348. IEEE, 2003.

DATTA, J. and CHAKRAVARTY, S. C. Chandrayaan-1. India's first scientific mission to Moon. *Oublications & Public Relations. Space Science Office. ISRO HQ, Bangalore*, 2004.

DEOLALIKAR, V., RECKER, J., MESARINA, M. and PRADHAN, S. *Optimal scheduling for networks of RFID readers Paper presented at the first international workshop on RFID and ubiquitous sensor networks*, 2005.

DHILLON, S., CHAKRABARTY, K. and IYENGAR, S. Sensor placement for grid coverage under imprecise detections. In *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, vol. 2, pages 1581–1587. IEEE, 2002.

DRUD, A. CONOPT: A GRG code for large sparse dynamic nonlinear optimization problems. *Mathematical Programming*, vol. 31(2), pages 153–191, 1985.

DUBOIS, P., BOTTERON, C., MITEV, V., MENON, C., FARINE, P., DAINESI, P. and IONESCU, A. Ad hoc wireless sensor networks for exploration of Solar-system bodies. *Acta Astronautica*, vol. 64(5-6), pages 626–643, 2009.

DUBOWSKY, S., IAGNEMMA, K., LIBERATORE, S., LAMBETH, D., PLANTE, J. and BOSTON, P. A Concept Mission: Microbots for Large-Scale Planetary Surface and Subsurface Exploration. In *AIP Conference Proceedings*, vol. 746, page 1449. 2005.

DUCATEL, K., BOGDANOWICZ, M., SCAPOLO, F., LEIJTEN, J. and BURGELMAN, J.-C. Scenarios for Ambient Intelligence in 2010. Informe técnico, Information Society Technologies Advisory Group, 2001.

ELPAÍS.COM. Detenidos 12 cazadores furtivos en la Operación Bambi. 2008.

ELSON, J., GIROD, L. and ESTRIN, D. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, vol. 36(SI), pages 147–163, 2002.

E.N., G. Random plane networks. *Journal of the Society for Industrial and Applied Mathematicsl*, vol. 9(4), pages 533–543, 1961.

EOM, J.-B., LEE, T.-J., RIETMAN, R. and YENER, A. An efficient framed-slotted ALOHA algorithm with pilot frame and binary selection for anti-collision of RFID tags. *Communications Letters, IEEE*, vol. 12(11), pages 861–863, 2008. ISSN 1089-7798.

EOM, J.-B., YIM, S.-B. and LEE, T.-J. An Efficient Reader Anticollision Algorithm in Dense RFID Networks With Mobile RFID Readers. *Industrial Electronics, IEEE Transactions on*, vol. 56(7), pages 2326–2336, 2009. ISSN 0278-0046.

EPCGLOBAL. EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface. Protocol for Communications at 860 MHz 960 MHz Version 2.0.0. *EPC Global*, 2013.

FANG, L., DU, W. and NING, P. A beacon-less location discovery scheme for wireless sensor networks. *Secure localization and time synchronization for wireless sensor and ad hoc networks*, page 33, 2006.

FERRERO, R., GANDINO, F., MONTRUCCHIO, B. and REBAUDENGO, M. A Fair and High Throughput Reader-to-Reader Anticollision Protocol in Dense RFID Networks. *Industrial Informatics, IEEE Transactions on*, vol. 8(3), pages 697–706, 2012. ISSN 1551-3203.

FIX, E. and HODGES JR, J. L. Discriminatory analysis-nonparametric discrimination: consistency properties. 1951.

FLOERKEMEIER, C. and SARMA, S. RFIDSim 2014;A Physical and Logical Layer Simulation Engine for Passive RFID. *Automation Science and Engineering, IEEE Transactions on*, vol. 6(1), pages 33–43, 2009. ISSN 1545-5955.

GAETA, E., CEA, G., ARREDONDO, M.-T. and LEUTERITZ, J. P. AmIRTEM: A Functional Model for Training of Aerobic Endurance for Health Improvement. *Biomedical Engineering, IEEE Transactions on*, vol. 59(11), pages 3155–3161, 2012. ISSN 0018-9294.

GANDINO, F., FERRERO, R., MONTRUCCHIO, B. and REBAUDENGO, M. DCNS: An Adaptable High Throughput RFID Reader-to-Reader Anticollision Protocol. *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24(5), pages 893–905, 2013. ISSN 1045-9219.

GANERIWAL, S., KUMAR, R. and SRIVASTAVA, M. Timing-sync protocol for sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 138–149. ACM, 2003.

GARCÍA-PALOMARES, U., BURGUILLO-RIAL, J. and GONZÁLEZ-CASTAÑO, F. Explicit gradient information in multiobjective optimization. *Operations research letters*, vol. 36(6), pages 722–725, 2008.

GARCÍA-PALOMARES, U., GONZÁLEZ-CASTAÑO, F. and BURGUILLO-RIAL, J. A combined global & local search (CGLS) approach to global optimization. *Journal of Global Optimization*, vol. 34(3), pages 409–426, 2006.

GARCÍA-PALOMARES, U. and RODRÍGUEZ, J. New Sequential and Parallel Derivative-Free Algorithms for Unconstrained Minimization. *SIAM Journal on Optimization*, vol. 13, page 79, 2002.

GARG, L., MCCLEAN, S., BARTON, M., MEENAN, B. and FULLERTON, K. Intelligent Patient Management and Resource Planning for Complex, Heterogeneous, and Stochastic Healthcare Systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 42(6), pages 1332–1345, 2012. ISSN 1083-4427.

GAUKLER, G. Item-level RFID in a retail supply chain with stock-out-based substitution. *Industrial Informatics, IEEE Transactions on*, vol. 7(2), pages 362–370, 2011. ISSN 1551-3203.

GAURA, E. and NEWMAN, R. Wireless sensor networks: The quest for planetary field sensing. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 596–603. 2006.

GEORGE, S., APPLEBY, B. and FORTIER, L. *An application of a proposed airdrop planning system*. Proyecto Fin de Carrera, 2004.

GHASEMZADEH, H., LOSEU, V., GUENTERBERG, E. and JAFARI, R. Sport Training Using Body Sensor Networks: A Statistical Approach to Measure Wrist Rotation for Golf Swing. In *Proceedings of the Fourth International Conference on Body Area Networks*, page 2. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

GHOSH, A. and DAS, S. Coverage and connectivity issues in wireless sensor networks: A survey. *Pervasive and Mobile Computing*, vol. 4(3), pages 303–334, 2008.

GIROD, L. and ESTRIN, D. Robust range estimation using acoustic and multimodal sensing. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 3, pages 1312–1320. IEEE, 2001.

GONG, Y.-J., SHEN, M., ZHANG, J., KAYNAK, O., CHEN, W.-N. and ZHAN, Z.-H. Optimizing RFID Network Planning by Using a Particle Swarm Optimization Algorithm With Redundant Reader Elimination. *Industrial Informatics, IEEE Transactions on*, vol. 8(4), pages 900–912, 2012. ISSN 1551-3203.

GONZÁLEZ-CASTAÑO, F., COSTA-MONTENEGRO, E., BURGUILLO-RIAL, J. and GARCÍA-PALOMARES, U. Outdoor WLAN planning via non-monotone derivative-free optimization: algorithm adaptation and case study. *Computational Optimization and Applications*, vol. 40(3), pages 405–419, 2008.

GONZÁLEZ-CASTAÑO, F. J., VALES ALONSO, J., COSTA-MONTENEGRO, E., LÓPEZ-MATENCIO, P., VICENTE-CARRASCO, F., PARRADO-GARCÍA, F. J., GIL-CASTIÑEIRA, F. and COSTAS-RODRÍGUEZ, S. Acoustic Sensor Planning for Gunshot Location in National Parks: A Pareto Front Approach. *Sensors*, vol. 9(12), pages 9493–9512, 2009. ISSN 1424-8220.

GOWERS, T., BARROW-GREEN, J. and LEADER, I. *The Princeton companion to mathematics*. Princeton Univ Dept of Art &, 2008.

GUARDIA CIVIL, SPANISH MINISTRY OF INTERIOR. SEPRONA. Servicio de Protección a la Naturaleza. 2009.

HAILES, S. The SEnsing for Sport And Managed Exercise project (SESAME). 2006. Http://sesame-wiki.cl.cam.ac.uk/twiki/bin/view/Sesame (accessed on 17 March 2014).

HASKELL, W., LEE, I. ET AL. Physical activity and public health. Updated recommendation for adults from the American College of Sports Medicine and the American Heart Association. *Circulation*, 2007.

HAYAJNEH, T. and KHASAWNEH, S. Analysis and evaluation of random placement strategies in wireless sensor networks. *Journal of Circuits, Systems and Computers*, vol. 23(10), page 1450138, 2014.

HEINZ, E., KUNZE, K., GRUBER, M., BANNACH, D. and LUKOWICZ, P. Using Wearable Sensors for Real-Time Recognition Tasks in Games of Martial Arts - An Initial Experiment. In *Computational Intelligence and Games, 2006 IEEE Symposium on*, pages 98–102. 2006.

HEMMATI, H., SENGUPTA, A., CASTILLO, J., MCELRATH, T., ROBERTS, T. and WILLIS, P. Two-Dimensional Planetary Surface Landers. In *NASA Innovative Advanced Concepts (NIAC) Symposium*. 2014.

HUANG, C. and TSENG, Y. The coverage problem in a wireless sensor network. *Mobile Networks and Applications*, vol. 10(4), pages 519–528, 2005.

ISHIZUKA, M. and AIDA, M. Performance study of node placement in sensor networks. In *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on*, pages 598–603. IEEE, 2004.

ISLER, V., KANNAN, S. and DANIILIDIS, K. Sampling based sensor-network deployment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2004.

ISO. *ISO/IEC 18000-6C:2004: information technology - radio frequency identification for item management-Part 6: parameters for air interface communications at 860 MHz to 960 MHz*, 2010a.

ISO. *ISO/IEC 18000-7:2008: information technology - radio frequency identification for item management - Part 7: parameters for active air interface communications at 433 MHz*, 2010b.

JAYNES, E. and BRETTHORST, G. *Probability theory: the logic of science*. Cambridge Univ Pr, 2003.

JIN, M., RONG, G., WU, H., SHUAI, L. and GUO, X. Optimal surface deployment problem in wireless sensor networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 2345–2353. IEEE, 2012.

KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, vol. 82(Series D), pages 35–45, 1960.

KAMENETSKY, M. and UNBEHAUN, M. Coverage planning for outdoor wireless LAN systems. In *Broadband Communications, 2002. Access, Transmission, Networking. 2002 International Zurich Seminar on*, pages 49–1. IEEE, 2002.

KHANDELWAL, G., LEE, K., YENER, A. and SERBETLI, S. ASAP: A MAC Protocol for Dense and Time-constrained RFID Systems. *EURASIP J. Wirel. Commun. Netw.*, vol. 2007(2), pages 3–3, 2007. ISSN 1687-1472.

KIM, J., LEE, W., KIM, E., KIM, D. and SUH, K. Optimized transmission power control of interrogators for collision arbitration in UHF RFID systems. IEEE Commun. Lett. vol. 11(1), pages 22–24, 2007.

KINDERMANN, W., SIMON, G. and KEUL, J. The significance of the aerobic-anaerobic transition for the determination of work load intensities during endurance training. *European journal of applied physiology and occupational physiology*, vol. 42(1), pages 25–34, 1979.

KIRKPATRICK, S., GELATT, C. D. and VECCHI, M. P. Optimization by Simulated Annealing. *Science, Number 4598*, vol. 220, 4598, pages 671–680, 1983.

KRAUSE, A., SINGH, A. and GUESTRIN, C. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, vol. 9, pages 235–284, 2008.

KRUPENIO, N. Radar investigations of the moon. 1971.

LABORATORY, D. NASA Taps Draper for Low Cost Concept that Could Accelerate Planetary Exploration Using Cold Atom Technology, ChipSats. 2014.

LAND, A. and DOIG, A. An Automatic Method for Solving Discrete Programming Problems. In *50 Years of Integer Programming 1958-2008* (edited by M. JÃ$\frac{1}{4}$nger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi and L. A. Wolsey), pages 105–132. Springer Berlin Heidelberg, 2010. ISBN 978-3-540-68274-5.

LAURENT, A. Bombing problems-a statistical approach. *Operations Research*, pages 75–89, 1957.

LEE, S.-R., JOO, S.-D. and LEE, C.-W. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification. In *Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on*, pages 166–172. 2005.

LEONCINI, M., RESTA, G. and SANTI, P. Partially controlled deployment strategies for wireless sensors. *Ad Hoc Networks*, vol. 7(1), pages 1–23, 2009.

LEONG, K. S., NG, M. L., GRASSO, A. and COLE, P. Synchronization of RFID readers for dense RFID reader environments. In *Applications and the Internet Workshops, 2006. SAINT Workshops 2006. International Symposium on*, pages 4 pp.–51. 2006.

LI, Z. and HE, C. Optimal scheduling-based {RFID} reader-to-reader collision avoidance method using artificial immune system. *Applied Soft Computing*, vol. 13(5), pages 2557 – 2568, 2013.

LI, Z.-H., ZHENG, X.-J. and GUAN, J.-W. Machine Learning and Cybernetics, 2009 International Conference on. vol. 1, pages 198–203. 2009.

LIESKA, K., LAITINEN, E. and LAHTEENMAKI, J. Radio coverage optimization with genetic algorithms. In *Personal, Indoor and Mobile Radio Communications, 1998. The Ninth IEEE International Symposium on*, vol. 1, pages 318–322. IEEE, 1998.

LIU, B. and TOWSLEY, D. In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on, title=A study of the coverage of large-scale sensor networks*, pages 475 – 483. 2004.

LIU, S. and LIN, Y. Grey Incidence Analysis. *Grey information: Theory and practical applications*, pages 85–138, 2006.

LLOSA, J., VILAJOSANA, I., VILAJOSANA, X., NAVARRO, N., SURIÑACH, E. and MARQUÈS, J. M. REMOTE, a Wireless Sensor Network Based System to Monitor Rowing Performance. *Sensors*, vol. 9(9), pages 7069–7082, 2009. ISSN 1424-8220.

LOCATELLI, M. Simulated annealing algorithms for continuous global optimization: convergence conditions. *Journal of Optimization Theory and applications*, vol. 104(1), pages 121–133, 2000.

MAGRASSI, P. and BERG, T. A World of Smart Objects. In *Research report R-17-2243*. Gartner, 2002.

MAKONIN, S., BARTRAM, L. and POPOWICH, F. A Smarter Smart Home: Case Studies of Ambient Intelligence. *Pervasive Computing, IEEE*, vol. 12(1), pages 58–66, 2013. ISSN 1536-1268.

MANOBIANCO, J., CASE, J. L., EVANS, R. J., SHORT, D. A. and PISTER, K. S. Global Environmental MEMS Sensors (GEMS): A revolutionary observing system for the 21st century. *Phase I Final Report.[Available online at http://www. niac. usra. edu/studies/]*, 2002.

MCCLEESE, D., GREELEY, R. and MACPHERSON, G. Science planning for exploring mars. *Jet Propulsion Laboratory Publication 01-7*, 2001.

MCCORD, T. B., TAYLOR, L. A., COMBE, J.-P., KRAMER, G., PIETERS, C. M., SUNSHINE, J. M. and CLARK, R. N. Sources and physical processes responsible for OH/H2O in the lunar soil as revealed by the Moon Mineralogy Mapper (M3). *Journal of Geophysical Research: Planets*, vol. 116(E6), pages n/a–n/a, 2011. ISSN 2156-2202.

MEGUERDICHIAN, S., KOUSHANFAR, F., POTKONJAK, M. and SRIVASTAVA, M. Coverage problems in wireless ad-hoc sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pages 1380–1387 vol.3. 2001. ISSN 0743-166X.

MICHAHELLES, F. and SCHIELE, B. Sensing and monitoring professional skiers. *Pervasive Computing, IEEE*, vol. 4(3), pages 40–45, 2005.

MILENKOVIC, O. and COMPTON, K. Probabilistic transforms for combinatorial urn models. Comb. Probab.Comput. vol. 13(4-5), pages 645–675, 2004.

MIORANDI, D., SICARI, S., PELLEGRINI, F. D. and CHLAMTAC, I. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, vol. 10(7), pages 1497 – 1516, 2012. ISSN 1570-8705.

MOHSENIAN-RAD, A.-H., SHAH-MANSOURI, V., WONG, V. and SCHOBER, R. Distributed channel selection and randomized interrogation algorithms for large-scale and dense RFID systems. *Wireless Communications, IEEE Transactions on*, vol. 9(4), pages 1402–1413, 2010. ISSN 1536-1276.

MORAS, G. and ZURITA, C. Training intensity valoration through the cardiac frequency in volleyball. *Apunts PHYSICAL EDUCATION AND SPORTS*, (55), pages 77–84, 1999.

MORAS FELIU, G. *Complete Volleyball Training 1000 Exercises And Games*. Paidotribo, 1994.

NELLES, O. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer, 2001.

NEWS HOY.ES. Desarticulada un red de cazadores furtivos en la Sierra de Gredos. 2009.

NGUYEN, S. D., PHAM, T. T., FRIBOURG BLANC, E., LE, N. N., DANG, C. M. and TEDJINI, S. Approach for quality detection of food by RFID-based wireless sensor tag. *Electronics Letters*, vol. 49(25), pages 1588–1589, 2013. ISSN 0013-5194.

NURGE, M. A. In situ dielectric spectroscopy for water detection on the lunar surface . *Planetary and Space Science*, vol. 65(1), pages 76 – 82, 2012. ISSN 0032-0633.

OIKONOMOPOULOS, A., PATRAS, I. and PANTIC, M. Spatiotemporal salient points for visual recognition of human actions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36(3), pages 710–719, 2005. ISSN 1083-4419.

ONDEMIR, O., ILGIN, M. and GUPTA, S. Optimal End-of-Life Management in Closed-Loop Supply Chains Using RFID and Sensors. *Industrial Informatics, IEEE Transactions on*, vol. 8(3), pages 719–728, 2012. ISSN 1551-3203.

PÄÄKKÖNEN. Finnish Suppressor Project. 2012.

PABARI, J., ACHARYA, Y., DESAI, U. and MERCHANT, S. Concept of wireless sensor network for future in-situ exploration of lunar ice using wireless impedance sensor. *Advances in Space Research*, vol. 52(2), pages 321 – 331, 2013. ISSN 0273-1177. Lunar Exploration - {II}.

PABARI, J. P., ACHARYA, Y. B., DESAI, U. B., MERCHANT, S. N. and KRISHNA, B. G. Radio Frequency Modelling for Future Wireless Sensor Network on Surface of the Moon. *IJCNS*, vol. 3(4), pages 395–401, 2010.

PARRADO-GARCÍA, F., LÓPEZ-MATENCIO, P., CHAVES-DIÉGUEZ, D., VALES-ALONSO, J., ALCARAZ, J. and GONZÁLEZ-CASTAÑO, F. Evaluation of Team-Sport Training Effort Control Systems. In *Human-Computer Systems Interaction: Backgrounds and Applications 3* (edited by Z. S. Hippe, J. L. Kulikowski, T. Mroczek and J. Wtorek), vol. 300 of *Advances in Intelligent Systems and Computing*, pages 337–355. Springer International Publishing, 2014. ISBN 978-3-319-08490-9.

PATEK, S., LOGAN, D. and CASTANON, D. Approximate dynamic programming for the solution of multiplatform path planning problems. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, vol. 1, pages 1061–1066 vol.1. 1999. ISSN 1062-922X.

PATWARI, N. *Location estimation in sensor networks*. PhD Thesis, Citeseer, 2005.

PENROSE, M. *Random geometric graphs*, vol. 5. Oxford University Press Oxford, 2003.

PFISTERER, D., LIPPHARDT, M., BUSCHMANN, C., HELLBRUECK, H., FISCHER, S. and SAUSELIN, J. MarathonNet: Adding value to large scale sport events-a connectivity analysis. In *Proceedings of the First International Conference on Integrated Internet Ad Hoc and Sensor Networks*, page 12. ACM, 2006.

PIETERS, C. M., GOSWAMI, J. N., CLARK, R. N., ANNADURAI, M., BOARDMAN, J., BURATTI, B., COMBE, J.-P., DYAR, M. D., GREEN, R., HEAD, J. W., HIBBITTS, C., HICKS, M., ISAACSON, P., KLIMA, R., KRAMER, G., KUMAR, S., LIVO, E., LUNDEEN, S., MALARET, E., MCCORD, T., MUSTARD, J., NETTLES, J., PETRO, N., RUNYON, C., STAID, M., SUNSHINE, J., TAYLOR, L. A., TOMPKINS, S. and VARANASI, P. Character and Spatial Distribution of OH/H2O on the Surface of the Moon Seen by M3 on Chandrayaan-1. *Science*, vol. 326(5952), pages 568–572, 2009.

PILARZYK, J. Lithium Carbon Monofluoride Coin Cells in Real-Time Clock and Memory Backup Applications. *Rayovac, White Papers*, 1997.

PODURI, S. and SUKHATME, G. Constrained coverage for mobile sensor networks. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 1, pages 165–171 Vol.1. 2004. ISSN 1050-4729.

POLAR. Team sports products from Polar Company. 2011. Http://www.polarusa.com/us-en/b2b_products/team_sports/polar_team2_pro (accessed on 17 March 2014).

POMARICO-FRANQUIZ, J. and SHMALIY, Y. Accurate Self-Localization in RFID Tag Information Grids Using FIR Filtering. *Industrial Informatics, IEEE Transactions on*, vol. 10(2), pages 1317–1326, 2014. ISSN 1551-3203.

PRASAD, K. D. and MURTY, S. Wireless sensor networks - a potential tool to probe for water on moon. *Advances in Space Research*, vol. 48(3), pages 601 – 612, 2011. ISSN 0273-1177.

R. KLIMA, J. H., J. CAHILL and LAWRENCE, D. Remote detection of magmatic water in Bullialdus Crater on the Moon. 2013.

REASON, J. and CREPALDI, R. Ambient intelligence for freight railroads. *IBM Journal of Research and Development*, vol. 53(3), pages 6:1–6:14, 2009. ISSN 0018-8646.

RIENER, A., FULLERTON, M., MAAG, C., MARK, C., BELTRAN RUIZ, C., MINGUEZ RUBIO, J. and ZIA, K. Modular Simulation-Based Physical and Emotional Assessment of Ambient Intelligence in Traffic. *Human-Machine Systems, IEEE Transactions on*, vol. 44(2), pages 286–292, 2014. ISSN 2168-2291.

ROBERGS, R. A. and LANDWEHR, R. The surprising history of the "HRmax= 220-age" equation. *J Exerc Physiol*, vol. 5(2), pages 1–10, 2002.

SAASTAMOINEN, K., KETOLA, J. and TURUNEN, E. Defining athlete's anaerobic and aerobic thresholds by using similarity measures and differential evolution. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 2, pages 1331–1335 vol.2. 2004. ISSN 1062-922X.

SEO, H. and LEE, C. A New GA-Based Resource Allocation Scheme for a Reader-to-Reader Interference Problem in RFID Systems. *Paper presented in 2010 IEEE international conference on communications (ICC)*, 2010.

SEVGI, C. and KOÇYIGIT, A. Optimal deployment in randomly deployed heterogeneous WSNs: A connected coverage approach. *Journal of Network and Computer Applications*, vol. 46(0), pages 182 – 197, 2014. ISSN 1084-8045.

SHAH-MANSOURI, V. and WONG, V. Cardinality Estimation in RFID Systems with Multiple Readers. *Wireless Communications, IEEE Transactions on*, vol. 10(5), pages 1458–1469, 2011. ISSN 1536-1276.

SHAHZAD, M. and LIU, A. Fast and Accurate Estimation of RFID Tags. *Networking, IEEE/ACM Transactions on*, vol. PP(99), pages 1–1, 2014. ISSN 1063-6692.

SHAO, L., ZHEN, X., TAO, D. and LI, X. Spatio-Temporal Laplacian Pyramid Coding for Action Recognition. *Cybernetics, IEEE Transactions on*, vol. PP(99), pages 1–1, 2014. ISSN 2168-2267.

SHERALI, H., PENDYALA, C. and RAPPAPORT, T. Optimal location of transmitters for micro-cellular radio communication system design. *Selected Areas in Communications, IEEE Journal on*, vol. 14(4), pages 662–673, 1996.

SIPPER, M., AZARIA, Y., HAUPTMAN, A. and SHICHEL, Y. Designing an Evolutionary Strategizing Machine for Game Playing and Beyond. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37(4), pages 583–593, 2007. ISSN 1094-6977.

SPANISH GOVERMENT. Spanish National Parks - Cabañeros. 2004.

SPELMEZAN, D. and BORCHERS, J. Chi'08 extended abstracts on human factors in computing systems. pages 3327–3332. ACM, 2008.

SPIRIT. Mars Exploration Rover Mission: Home. 2011.

STIMPSON, A. and CUMMINGS, M. Assessing Intervention Timing in Computer-Based Education Using Machine Learning Algorithms. *Access, IEEE*, vol. 2, pages 78–87, 2014. ISSN 2169-3536.

SUNSHINE, J. M., FARNHAM, T. L., FEAGA, L. M., GROUSSIN, O., MERLIN, F., MILLIKEN, R. E. and AHEARN, M. F. Temporal and Spatial Variability of Lunar Hydration As Observed by the Deep Impact Spacecraft. *Science*, vol. 326(5952), pages 565–568, 2009.

SUYOUNG, Y. and SICHITIU, M. L. Analysis and performance evaluation of a time synchronization protocol for wireless sensor networks. 2005.

TANAKA, H., MONAHAN, K. and SEALS, D. Age-predicted maximal heart rate revisited. *Journal of the American College of Cardiology*, vol. 37(1), page 153, 2001.

TANAKA, Y. and SASASE, I. Interference avoidance algorithms for passive RFID systems using contention-based transmit abortion. IEICE Trans. Commun. vol. E90-B(11), pages 3170–3180, 2007.

TEMEL, S., UNALDI, N. and KAYNAK, O. On Deployment of Wireless Sensors on 3-D Terrains to Maximize Sensing Coverage by Utilizing Cat Swarm Optimization With Wavelet Transform. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 44(1), pages 111–120, 2014. ISSN 2168-2216.

UCAM. UCAM Murcia Volleyball Team. 2014. Http://www.voleymurcia.es.

UNBEHAUN, M. and KAMENETSKY, M. On the deployment of picocellular wireless infrastructure. *Wireless Communications, IEEE*, vol. 10(6), pages 70–80, 2003.

VALES-ALONSO, J., BUENO-DELGADO, V., EGEA-LOPEZ, E., GONZALEZ-CASTANO, F. and ALCARAZ, J. Multiframe Maximum-Likelihood Tag Estimation for RFID Anticollision Protocols. *Industrial Informatics, IEEE Transactions on*, vol. 7(3), pages 487–496, 2011. ISSN 1551-3203.

VALES-ALONSO, J., CHAVES-DIEGUEZ, D., LOPEZ-MATENCIO, P., ALCARAZ, J., PARRADO-GARCIA, F. and GONZALEZ-CASTANO, F. SAETA: A Smart Coaching Assistant for Professional Volleyball Training. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. PP(99), pages 1–1, 2015. ISSN 2168-2216.

VALES-ALONSO, J., EGEA-LÓPEZ, E., MARTÍNEZ-SALA, A., PAVÓN-MERIÑO, P., BUENO-DELGADO, M. V. and GARCÍA-HARO, J. Performance evaluation of MAC transmission power control in wireless sensor networks. *Computer Networks*, vol. 51(6), pages 1483 – 1498, 2007.

VALES-ALONSO, J., FERNÁNDEZ, J., GONZÁLEZ-CASTAÑO, F. J. and CABALLERO, A. A parallel optimization approach for controlling allele diversity in conservation schemes. 2003.

VALES-ALONSO, J., LÓPEZ-MATENCIO, P., ALCARAZ, J., SIEIRO-LOMBA, J., COSTA-MONTENEGRO, E. and GONZÁLEZ-CASTAÑO, F. A Dynamic Programming Approach for Ambient Intelligence Platforms in Running Sports Based on Markov Decision Processes. *Human–Computer Systems Interaction: Backgrounds and Applications 2*, pages 165–181, 2012a.

VALES-ALONSO, J., LÓPEZ-MATENCIO, P., GONZALEZ-CASTAÑO, F., NAVARRO-HELLÍN, H., BAÑOS-GUIRAO, P., PÉREZ-MARTÍNEZ, F., MARTÍNEZ-ÁLVAREZ, R., GONZÁLEZ-JIMÉNEZ, D., GIL-CASTIÑEIRA, F. and DURO-FERNÁNDEZ, R. Ambient Intelligence Systems for Personalized Sport Training. *Sensors*, vol. 10(3), pages 2359–2385, 2010. ISSN 1424-8220.

VALES-ALONSO, J., LOPEZ-MATENCIO, P., VEIGA-GONTAN, J., GUIRAO, P. B. and ALCARAZ, J. An effort control system for training elite team-sport athletes. In *Human System Interaction (HSI), 2013 The 6th International Conference on*, pages 279–286. IEEE, 2013a.

VALES-ALONSO, J., PARRADO-GARCÍA, F., LÓPEZ-MATENCIO, P., ALCARAZ, J. and GONZÁLEZ-CASTAÑO, F. On the optimal random deployment of wireless sensor networks in non-homogeneous scenarios. *Ad Hoc Networks*, vol. 11(3), pages 846 – 860, 2013b. ISSN 1570-8705.

VALES-ALONSO, J., PARRADO-GARCÍA, F. and ALCARAZ, J. Analytical Computation of the Mean Number of Tag Identifications During a Time Interval in FSA. *Communications Letters, IEEE*, vol. 18(11), pages 1923–1926, 2014. ISSN 1089-7798.

VALES-ALONSO, J., PARRADO-GARCÍA, F., ALCARAZ, J. and EGEA-LOPEZ, E. Optimal scheduling in single channel dense reader RFID environments. *Paper presented in 2012 fourth international EURASIP workshop on RFID technology (EURASIP RFID)*, 2012b.

VALES-ALONSO, J., PARRADO-GARCÍA, F. J. and ALCARAZ, J. Performance analysis of optimal schedulers in single channel dense radio frequency identification environments. *EURASIP Journal on Embedded Systems*, vol. 2013(1), page 11, 2013c. ISSN 1687-3963.

VALES-ALONSO, J., PARRADO-GARCÍA, F. J. and ALCARAZ, J. J. Performance analysis of optimal schedulers in single channel dense radio frequency identification environments. *EURASIP Journal on Embedded Systems*, vol. 2013(1), 2013d.

VOGT, H. Efficient Object Identification with Passive RFID Tags. vol. 2414, pages 98–113, 2002.

WANG, D., XIE, B. and AGRAWAL, D. Coverage and lifetime optimization of wireless sensor networks with gaussian distribution. *IEEE transactions on mobile computing*, pages 1444–1458, 2008.

WANG, Q., XU, K., TAKAHARA, G. and HASSANEIN, H. WSN04-1: Deployment for Information Oriented Sensing Coverage in Wireless Sensor Networks.

In *Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE*, pages 1–5. IEEE, 2006.

WATANABE, K. and HOKARI, M. Kinematical analysis and measurement of sports form. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 36(3), pages 549–557, 2006. ISSN 1083-4427.

WESTERLUND, P. F., T. An Extended Cutting Plane (ECP) Method for the Solution of MINLP Problems. 1992.

WONG, K. Abstract Field Strength Prediction in Irregular Terrain-the PTP Model. 2002.

WRISBERG, C. *Sport skill instruction for coaches*. Human Kinetics Publishers, 2007.

WU, Y., SHENG, Q. Z., SHEN, H. and ZEADALLY, S. Modeling Object Flows from Distributed and Federated RFID Data Streams for Efficient Tracking and Tracing. *IEEE Trans. Parallel Distrib. Syst.*, vol. 24(10), pages 2036–2045, 2013. ISSN 1045-9219.

XHAFA, F., CABALLE, S., BESSIS, N., JUAN, A., BAROLLI, L. and MIHO, R. Using Massive Processing and Mining for Modelling and Decision Making in Online Learning Systems. In *Emerging Intelligent Data and Web Technologies (EIDWT), 2011 International Conference on*, pages 91–98. 2011.

XU, K., HASSANEIN, H., TAKAHARA, G. and WANG, Q. Relay node deployment strategies in heterogeneous wireless sensor networks. *Mobile Computing, IEEE Transactions on*, vol. 9(2), pages 145–159, 2010.

XU, L., HE, W. and LI, S. Internet of Things in Industries: A Survey. 2014.

YANG, Q. and CHEN, W. Grey incidence analysis on influence factors of university volleyball teams' performance. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 1–5. 2012.

YOUNIS, M. and AKKAYA, K. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks*, vol. 6(4), pages 621 – 655, 2008.

ZAMORA-IZQUIERDO, M., SANTA, J. and GOÓMEZ-SKARMETA, A. An Integral and Networked Home Automation Solution for Indoor Ambient Intelligence. *Pervasive Computing, IEEE*, vol. 9(4), pages 66–77, 2010. ISSN 1536-1268.

ZHANG, C. and WONG, A. K. C. Toward efficient multiple molecular sequence alignment: a system of genetic algorithm and dynamic programming. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 27(6), pages 918–932, 1997. ISSN 1083-4419.

ZHANG, J., LUO, J. and LUO, X. A Robust Localization Algorithm for Wireless Sensor Networks. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on*, pages 1–4. IEEE, 2008.

ZHU, L. and YUM, T. Optimal Framed Aloha Based Anti-Collision Algorithms for RFID Systems. *Communications, IEEE Transactions on*, vol. 58(12), pages 3583–3592, 2010. ISSN 0090-6778.

ZINTL, F. *Endurance Training: Foundation, Methods and Training Management*. Ediciones Martínez Roca, 1991.

ZOU, Y. and CHAKRABARTY, K. Uncertainty-aware sensor deployment algorithms for surveillance applications. *GLOBECOM-NEW YORK-*, vol. 5, pages 2972–2976, 2003.

– And where does the newborn go from here?
– The net is vast and infinite.

Tenía que hacerlo ¿entiende?
¡Tenía que hacerlo!